ATARI

ST COMPUTER

Die Fachzeitschrift für den ATARI-ST Anwender.

Juli / August 89

DM 7,- Ös. 56,- Sfr. 7,-



PC-Speed

Hardware-MS-DOS-Emulator

Statistik auf dem ST

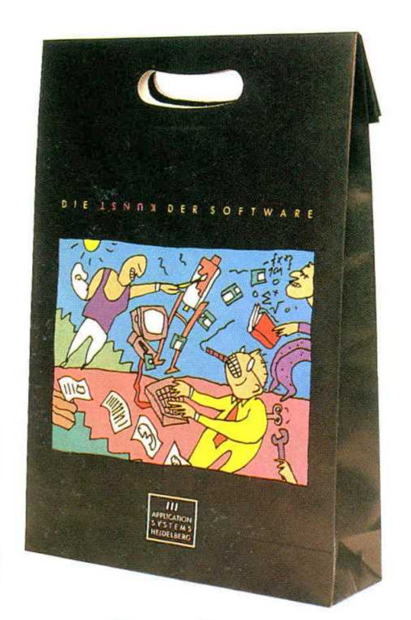
16 MHz-Takt

Doppelt so schnell?

1.2 MB-Disketten im ATARI-Laufwerk

gw

tüte.



Wo etwas verkauft wird, da sind auch Tüten und niemand würde in einer Anzeige ein Wort darüber verlieren. Aber wir finden unsere Tüte so schön, daß wir sie oben abbilden, und wir sind ein bißchen stolz, weil es unsere erste Tüte ist. Man kann diese Tüte an unserem Stand auf der Atari-Messe in Düsseldorf bekommen (25.-27.August 1989) oder bei ausgewählten Händlern, die unsere Software führen. Natürlich nur, wenn man etwas kauft. Ahem.



Tüte: Find 'ich gut.

Application Systems Heidelberg, Postfach 10 26 46, D-6900 Heidelberg 1

EDITORIAL

Emulationen

Seit der ATARI ST auf den Markt gekommen ist, hat sich herausgestellt, daß die ST-Besitzer immer noch weitere und bessere Features aus ihrem Rechner herausholen wollen. Gerade auf dem ST ist die Zahl der Programmierer ziemlich hoch, die dabei auch die Grenzen zwischen dem ST und anderen Rechnern geradezu mühelos aus dem Wege räumen und sogenannte Emulatoren schreiben. Mit diesen werden fremde Rechner - also keine ATARI STs - so nachempfunden, daß ein Großteil der für den anderen Computer geschriebenen Software läuft.

Allerdings darf man nie vergessen, daß ein Emulator keinesfalls ein vollwertiger Ersatz für einen Original-Rechner ist. Man bedenke nur, daß noch nicht einmal die "IBM-kompatiblen" Rechner so kompatibel sind, daß man mit ruhigen Gewissen sagen könnte, kein Zweifel 100% Kompatibilität. Bei Emulatoren muß - selbst bei gleichen Prozessoren - immer die unterschiedliche übrige Hardware softwaremäßig ersetzt werden, was zumindest immer ein Zeitverlust bedeutet (von irgendwelchen fehlenden Sound- und Grafikfähigkeiten mal ganz zu schweigen). Sie sind somit nur ein Ersatz für entweder teurere Systeme (man denke da z.B. an den Macintosh) oder werden einfach mal zum reinschnuppern und aus Platzmangel gekauft, da nicht jeder den Platz für zwei, drei Computer in seinen vier Wänden hat. Viele Anwender arbeiten auch im Beruf an Originalrechnern und bringen dann ihre Arbeit mit nach Hause, denn meistens kriegt man dann die besten Ideen, wenn weit und breit kein Rechner steht.

Man darf nie vergessen, daß ein Emulator keinesfalls ein vollwertiger Ersatz für einen Original-Rechner ist.

Um nun auch dem nicht geringen Teil der Emulator-Anwender auf dem ATARI ST gerecht zu werden, haben wir ab dieser Ausgabe eine Rubrik in der ST Computer eingeführt, die sich mit neuen Produkten und Tips für die Emulatoren beschäftigt. Sie sind natürlich auch aufgefordert, sich rege an dieser Rubrik zu beteiligen.

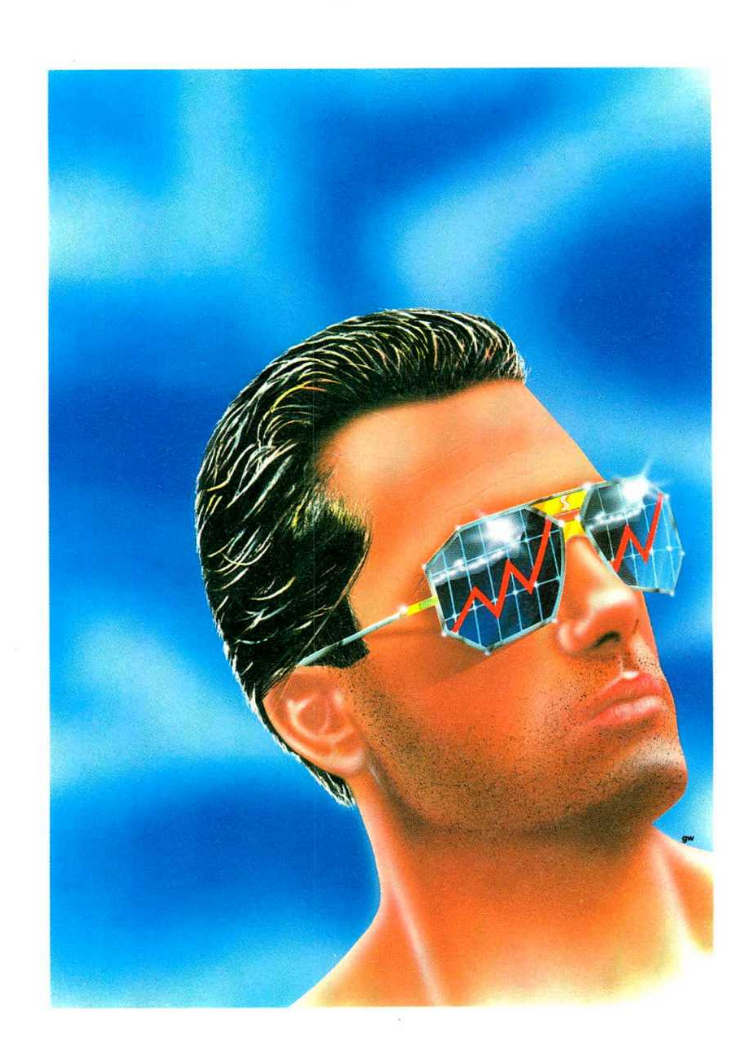
Harald Egel

INHALT

SOFTWARE

HARDWARE

GRUNDLAGEN



Wissenschaftliche Statistik

Abgesehen von Statistikpaketen, die unter dem Betriebssystem MS-DOS laufen und nicht selten die Eintausend-Mark-Grenze weit überschreiten, wurden für den ATARI ST in den letzten 2 Jahren einige Statistikpakete entwickelt bzw. adaptiert, die vom Preis-/Leistungsverhältnis her äußerst attraktiv sind. Wir haben uns vier dieser Pakete für Sie angesehen.

Seite **47**

Dem Floppy-Controller Dampf gemacht

1.2 MB auf einer Standarddiskette

In diesem Artikel wird von einer bisher beim ATARI ST nicht genutzten Möglichkeit die Rede sein, wie man noch mehr Daten auf seinen Standarddisketten unterbringen kann. Bekannt ist bislang nur das sogenannte Höherformatieren, neu ist dagegen wie man 1.2 MB auf einer Standarddiskette unterbringt.

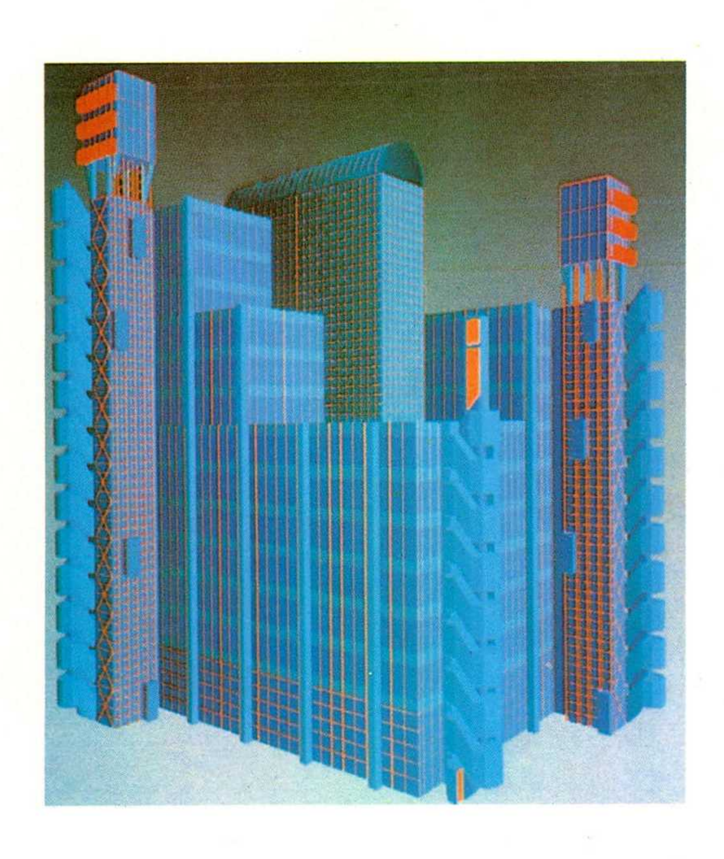
Seite **131**

PC-Speed

Hardware-MS-DOS-Emulator

ATARI hatte schon vor einigen Jahren einen Hardware-MS-DOS-Emulator angekündigt, der dann nie in den Handel kam. Auch der sogenannte Supercharger kam nie auf den Markt. Ein dritter Emulator stellte sich auch als Flopp heraus. So liest sich die Geschichte der Hardware-MS-DOS-Emulatoren für den ATARI ST. Diesen Sommer sollen nun gleich zwei neue Produkte auf den Markt kommen, PC-Speed und PC ditto II. Wir stellen Ihnen heute PC-Speed vor. Lassen Sie sich überraschen.

Seite **15**



Axis

3D-Zeichnungen & Animationen

Erste Versionen von Axis waren bereits auf der ATARI-Messe in Düsseldorf im Jahre 1987 zu sehen. Endgültig vorgestellt wurde das Produkt des holländischen Herstellers Modern Medium aber erst auf der diesjährigen CeBIT; eine Vorstellung, die durch eine auf Videokassette verfügbare und auch verhältnismäßig lange Animation recht eindrucksvoll war. Lesen Sie mehr darüber auf...

Seite **42**

PROGRAMMIERPRAXIS

EMULATIONEN

ANWENDUNGEN

AKTUELLES

RUBRIKEN

NEWS

ATARI-Messe Düsseldorf

Zum dritten Mal veranstaltet ATARI eine eigene Verkaufsmesse auf dem Düsseldorfer Messegelände. Durch den Erfolg der letzten zwei Jahre angespornt, hat ATARI die Ausstellungsfläche in diesem Jahr um 50 Prozent gegenüber dem Vorjahr auf 12000 qm erhöht. Erwartet werden rund 30.000 Besucher, womit die magische und höchstzulässige Besucherzahl für die beiden Messehallen erreicht wäre. In der Halle 1 und 2 werden ATARI und Aussteller aus dem In- und Ausland Neuigkeiten und Altbewährtes rund um die ATARI Computer präsentieren und zum Kauf anbieten; denn wie jedes Jahr ist diese Verkaufsmesse ein idealer Treffpunkt, um so manche Hard- und Software oder Zubehör gleich mit nach Hause zu nehmen. Natürlich ist auch in diesem Jahr der ATARI ST der Schwerpunkt der Veranstaltung, die am letzten Wochenende im August vom 25. bis einschließlich 27.08.89 stattfindet. Die Öffnungszeiten sind jeweils von 1000 bis 1800 Uhr und der Eintrittspreis beträgt DM 7,- pro Person für Schüler und Studenten DM 4,-. Die Eintrittskarten gelten auch für die Familienausstellung "Aktiv Leben '89", die gleichzeitig in den Hallen 3 bis 6 stattfindet. Ferner ist im Preis die Rückfahrt mit öffentlichen Verkehrsmitteln bis VRR Stufe 5 enthalten.

Was wird geboten?

ATARI wird das neue TT Modell zum erstenmal der Öffentlichkeit vorstellen, man darf gespannt sein. Im PC-Bereich wird der Portfolio, der PC im Taschenformat, sowie ein Low-Cost-Produkt der AT-Klasse zu sehen sein. In beiden Hallen haben Besucher je nach Interessensgebiet die Möglichkeit, in einem speziellen Forum neue Produkte kennen zu lernen und an Podiumsdiskussionen teilzunehmen. Der Rahmen reicht von der Schulsoftware über professionelle Branchenlösungen bis hin zu industriellen Lösungsbeispielen. Dem Midi-Bereich wird ein besonders großer Rahmen in Halle 1 gewidmet. Geschäftsführer, Alwin Stumpf, freut sich ein neues Midi-Keyboard die sogenennte 'Hotz-Box' (wir berichteten schon in unserer letzten Ausgabe darüber) vorstellen zu können.

Natürlich ist auch MAXON Computer mit einem großen Stand in Halle 2 vertreten, so daß Sie sich mit Fragen und Anregungen direkt an die Redaktion der ST Computer wenden können. Als besondere Überaschung werden wir unsere neue Grafikkarte MGE für alle Mega STs vorführen. Diesen Augenschmaus sollten Sie sich nicht entgehen lassen! Ferner stellen wir zwei neue Bücher vor, die die Themen "Datenstrukturen" und "Computersimulationen" behandeln.

Zuletzt sollten Sie sich auch den neuen Hardware-MS-DOS-Emulator nicht entgehen lassen.

Also bis bald in Düsseldorf

Zeichnen mit Arabesque

Mit Arabesque ist ein weiteres Grafikprogramm für den ATARI ST auf den Markt gekommen. Das Besondere an ihm ist, daß mit ihm sowohl Raster als auch Vektorgrafiken bearbeitet werden können. Als 'digitaler Zeichenstift' ermöglicht es die Verwendung umfangreicher Funktionen in mehreren Vergrößerungsstufen auf parkatisch beliebig großen Seiten. Die dabei in der Rastergrafik entstandenen Bilder lassen sich auf vielfache Art und Weise bearbeiten. Schwertpunkte bei der Vektorgrafik sind u.a. die hohe Auflösung von 11520x17280 Punkten (bei DIN A4) und somit entsprechende Zeichengenauigkeit, formatierter Text in voller Druckerauflösung und gute Ausdrucke. Es läßt sich Rastergrafik als Objekt in die Vektorgrafik übernehmen oder Vektorgrafik direkt in eine Rastergrafik zeichnen. Umfangreiche Druckoptionen für Laserdrucker und 24-Nadler sind vorhanden. Die Druckertreiber lassen sich edieren. Ausgeliefert wird Arabesque im Schuber und mit 200seitigen Handbuch. Das Programm benötigt mindestens 1 MB RAM, eine S/W-Monitor und ein doppelseitiges Diskettenlaufwerk. Der Preis beträgt DM 278,-.

SHIFT
Sonnenschein und Hansen
Unterer Lautrupweg 8
2390 Flensburg
Tel.: 0461/22828

Eickmann EX 110 & neue EX 60 L

Jetzt leiser!

Die Firma Eickmann Computer aus Frankfurt reagierte sofort auf den Test in der ST Computer vom Juni 89. Hier fiel die durch das Miniscribe-Laufwerk bedingte Geräuschkulisse auf. Nach aktueller Information der Firma werden alle EX 110 ab sofort mit leiseren Seagate-Platten ausgerüstet, was den Geräuschpegel deutlich unter das Niveau der Megafile 60 senkt.

Zusätzlich bietet die Firma jetzt eine weitere 'superleise' Festplatte an. EX 60 L heißt das Kind, hat 60 MB Speicherkapazität, eine mittlere Zugriffszeit von 35 ms und kommt wie die EX 40 R ohne Lüfter aus.

Eickmann Computer
In der Römerstadt 249
6000 Frankfurt/Main 90
Tel.: 069/763409

DRAW! 3.0



Bei Draw! handelt es sich um ein pixelorientiertes Zeichenprogramm für den ST. Es arbeitet nur im Monochrommodus. Seine besonderen Stärken liegen in der der Zusammenarbeit mit Textprogrammen und der einfachen Bedienung. Es stehen dafür eine Vielzahl wohldurchdachter Funktionen zur Verfügung. Es verarbeitet GEM- und Signum!-Zeichensätze, kann Ecken abrunden, Kontraste verstärken und besitzt auch viele brauchbare Blockfunktionen wie z.B. gradweises Drehen, Verzerren etc. Ein ausführlicher Test folgt in der nächsten ST Computer.

Draw! 3.0 kostet DM 129,-; Besitzer einer alten Draw!-Version erhalten für DM 60,- ein Upgrade.

OMIKRON.Soft- & Hardware GmbH
Erlachstr. 15a
7534 Birkenfeld 2
Tel.: 07082/5386

BODONI 13 Bold	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
BODONI 14	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
BODONI 14 Bold	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
BODONI 16	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
BODONI 16 Bold	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
BODONI 22	ABCDEFGHIJKLMNOPQRSTUVWXYZ
CAPITALS	ABCDEFGHIJKLMNOPQRSTUVWXYZ

Bodoni-Layoutpaket für Signum!

Für jeden Liebhaber von Signum! gibt es jetzt ein professionelles Layoutpaket, das die komplette Schriftenfamilie Bodoni-Antiqua enthält. Auszeichnen tut sich die Bodoni-Schrift durch klares Schriftbild und gute Lesbarkeit. Geliefert werden drei Disketten mit acht verschiedenen Punktgrößen zwischen 7 und 16 Punkt. Darüberhinaus sind verschiedene Auszeichnungsschriften wie eine 22 Punkt Versalienschrift, diverse Fett- und Kursivschriften und zwei Initialiensätze zur Grafikeinbindung für Überschriften in beliebiger Größe enthalten. Alle europäischen Sonderzeichen sind entweder unmittelbar verfügbar oder durch Kombination leicht im Text zu erstellen.

Das Bodoni-Layoutpaket ist für 24-Nadeldrucker erhältlich. Eine Laserversion ist noch nicht lieferbar. Bis zum 15.7.89 ist das Paket zum Subskriptionspreis von DM 135,- zu beziehen, danach für DM 175,-.

Semiotic Soft
Dr. W.Scheppe / R. Hauber
Südl.Auffahrtsallee 22
8000 München 19
Tel.: 089/163130

Preiswertes Statistikprogramm

Mit ProSTatist stellt die Firma ST Profi-Partner aus Lübeck ein neues preiswertes Statistikprogramm vor, das sich ohne sich in Einzelheiten zu verlieren auf das Wesentliche beschränkt. In seiner ersten Version bietet es grundlegende Verfahren an, deren Aufzählung den hier zur Verfügung stehenden Rahmen sprengen würde. Es sind Standardschnittstellen zu den gängigen Text- und Grafikprogrammen sowie Datenbanken enthalten. Allgemeine Erweiterungen, die jederzeit möglich sind, hängen von der Resonanz der Anwenderschaft ab. Kundenspezifische Erweiterungen sind jederzeit gegen eine geringen Unkostenbeitrag möglich. Eine Demodiskette, die beim Kauf angerechnet wird, ist für DM 20,- erhältlich. Die Vollversion hat ein 100seitiges Handbuch und kostet DM 198,-.

ST Profi-Partner
Mönkhofer Weg 126
2400 Lübeck
Tel.: 0451/505367

MIDIMIX-Collection

Die Berliner Firma GEERDES hat ab sofort für alle ATARI-ST-Sequencer-Benutzer ein höchst interessantes Produkt in den Vertrieb genommen: die MIDIMIX Collection. Unter dem Motto: "Die interaktive CD" im 3,5"-Format sind bereits fast 100 Titel aus den Bereichen Internationale Hits & Oldies, Jazz, Europe Nationals, Klassik & Soundtracks lieferbar. Die Sequencer-Songdaten sind generell im Midi-File-Standardformat abgelegt, das inzwischen auch vom GCG-eigenen 1st Track-Sequencerprogramm gelesen werden kann. Zusätzlich sind die meisten Songs speziell auch für Twentyfour und Creator/Notator auf der Diskette vorhanden. Standardmäßig sind die Titel für das MT-32 arrangiert und können mit den üblichen Spurparametern des jeweils benutzten Sequencers leicht an jedes Equipment angepaßt werden. Jede Spur ist mit dem zugehörigen Instrument benannt. Auch die einzelnen Schlagzeuginstrumente belegen jeweils eigene Spuren.

Auf den Songdisketten befindet sich auch ein Accessory, mit dem gegebenfalls mitgelieferte Klänge in das MT-32 transferiert werden können. Auch ein Sichern der eigenen Daten ist möglich.

Die Midimix Collection wird ständig erweitert. Der Preis beträgt DM 35,- pro Diskette.

Eine musikalische Reise durch den vorderen Orient von der Türkei bis Indien mit 15 Beispielsongs, notiert, eingespielt und kommentiert ist ebenfalls zu erwerben. Der Preis beträgt DM 80,-. Diskette, Noten und Text sind im Lieferumfang enthalten.

Geerdes Midisystem Software
Bismarckstr. 84
1000 Berlin 12
Tel.: 030/316779

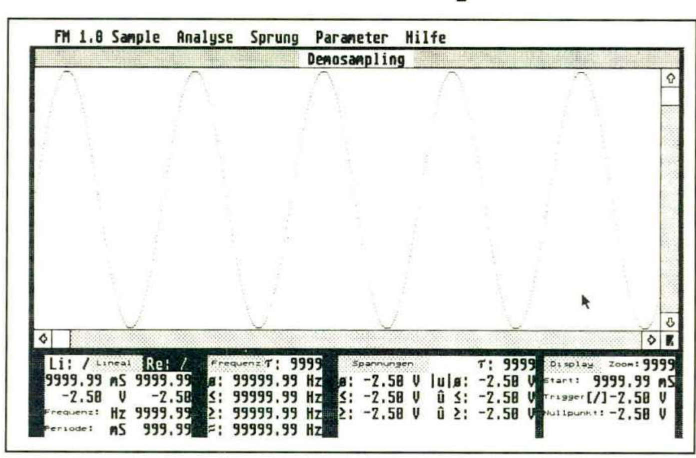
Belichtungsservice für Calamus

Am 1.Juli eröffnet der erste Belichtungsservice für Calamus in Frankfurt. Die im DTP-Bereich erfahrene Firma E.Gathof ermöglicht somit eine qualitativ hochwertige Ausgabe mit Calamus erstellter Dokumente. Die mit einer Linotronic 300 ausbelichteten Layouts haben dann eine maximale Auflösung von 2450 dpi. Die Ausgabe erfolgt je nach Wunsch auf Papier, Film oder direkt auf Offsetfilm. Jeder Calamus-Besitzer kann somit seine Layouts in Druckvorlagenqualität ausbelichten lassen. Preise und Modalitäten kann man unter folgender Adresse erfahren:

E.Gathof Belichtungs- &
Layoutfotosatzservice
Balduinstr. 80
6000 Frankfurt 70
Tel.: 069/655043 od. 653320

Neuartiges Meßtechnikpaket

Neu für den ATARI ST ist ein Meßtechnikpaket, das sich besonders zur Analyse niederfrequenter (bis etwa 50 kHz) Schwingungsverläufe sowie der automatisierten Erfassung und Protokollierung von Meßreihen eignet. Das Paket hat den Namen FM und besteht aus der Hardware des Soundsamplers Gadget sowie einer neuen Software. Einige Features sind:



- *8 Bit-Sampling-Hardware*
- *Abtastrate über 100 kHz*
- *Grafische Darstellung der gemessenen Spannungsverläufe*
- *Realtime-Berechnung von Spitzen- und Durchschnittswerten*
- *Protokollierung der Meßwerte und Übernahmemöglichkeit in andere Programme*
- *Auto-Modus für größere Meßreihen*

u.v.m.

FM ist auf allen STs lauffähig.

Das Paket kostet DM 298,- und ist unter folgender Adresse zu beziehen:

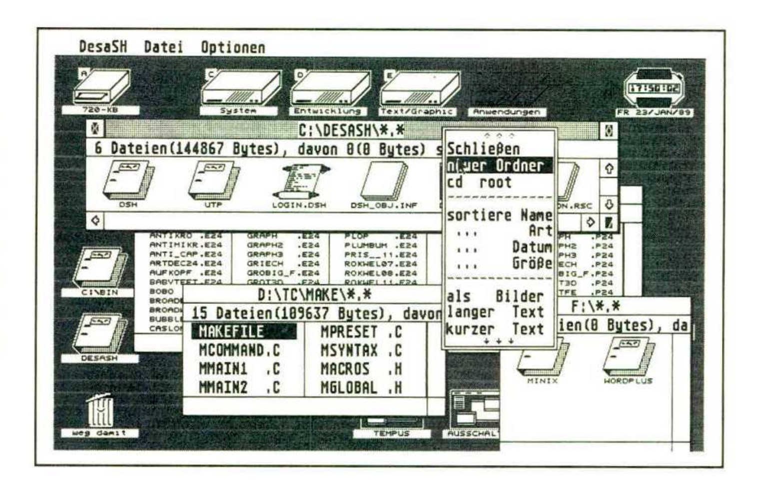
Sophisticated Applications
Computertechnik GbR
Friedrich-Ebert-Allee 2
2870 Delmenhorst
Tel.: 04221/14608

Echtzeituhr für den ROM-Port

Von Hard & Soft wird eine neue Echtzeituhr für den ROM-Port angeboten. Der Vorteil ist, daß kein Eingriff mehr in den Rechner notwendig ist. Die Uhr verfügt über eine Akkupufferung, eine hohe Ganggenauigkeit und Schaltjahrerkennung. Als Software wird 500 Bytes langer Uhrentreiber, ein Stellprogramm und ein Programm, daß Zeit und Datum in der Menüleiste anzeigt, mitgeliefert. Völlig neuartig ist das Start-UP-Modul, das für Anwender, die nur mit Disketten arbeiten, entwickelt wurde. Es erspart den dauernden Griff zur Bootdiskette, da es den Uhrentreiber in fremde Programme einbindet. Das Laden der Treibersoftware entfällt dadurch. Der Preis für das Uhrenmodul einschließlich Software beträgt DM 89,-.

Hard- & Software
A.Herberg
Bahnhofstr. 289
4620 Castrop-Rauxel
Tel.: 02305/12022

Neue Shell für den ST



Eine konzeptionell neue Benutzeroberfläche namens DesaShell wurde für den ATARI ST entwickelt. Sie umfaßt sämtliche gewohnten Desktop-Operationen, ist aber benutzerfreundlicher und verfügt über einen höheren Anwendungskomfort. DesaShell verbindet die Funktionen eines Command-Line-Interpreters mit denen des Desktops, wodurch man wahlweise mit dem einen als auch mit dem anderen arbeiten kann. Neben den gewohnten Features sind Batchdateien und beliebig viele unterschiedliche Icongruppen möglich. Die Icons können selber kreiert werden und sind veränder- und löschbar. Dateien lassen sich auf dem Desktop ablegen u.v.m.

Fahrig & Stamm
Rudolstädter Str. 14
1000 Berlin 31
Tel.: 030/8246749

14 Accessories

Neu aus den USA ist ein Produkt namens DeskCart!, das dort schon länger vertrieben wird und nun den Weg nach Deutschland gefunden hat. Es handelt sich dabei um ein Steckmodul, das in den ROM-Port gesteckt wird und dem Anwender u.a. jederzeit 14 Accessories zur Verfügung stellt. Ferner sind ein Kalender, eine Terminverwaltung, ein Notizbuch, ein Taschenrechner, ein Adressbuch usw. vorhanden. Das Programm liegt derzeit nur in englischer Fassung vor, eine deutsche Version ist allerdings in Vorbereitung. Es läuft sowohl in Schwarzweiß als auch in Farbe. Der Preis für die englische Version beträgt DM 185,-, für die deutsche DM 200,-.

Michael Weichselgartner
Chiemgaustr. 152
8000 München 90
Tel.: 089/6804642

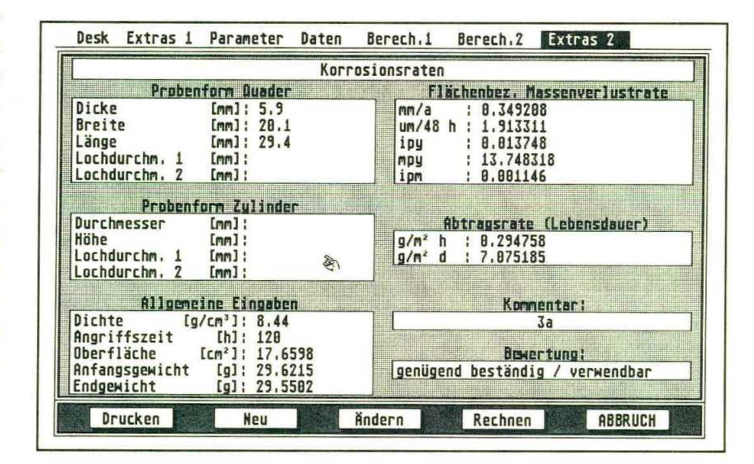
V_Manager 2.01

In einer neuen Version präsentiert sich das Versicherungsagenturprogramm V_Manager auf dem ATARI ST. Die neue Version ist gegenüber der alten erheblich erweitert worden. Der Maskenaufbau ist deutlich beschleunigt worden. Neu ist sind außerdem eine erweiterte Provisionsberechnung, eine Hilfefunktion, Datenimport und -export-Funktionen mit Datenbanken (z.B. dBase, Adimens), Verknüpfen von VM-Dateien, ein integrierter Texteditor angepaßt an den Listengenerator, ein großer Notizblock u.v.m. Der Preis von DM 1498,- hat sich nicht geändert. Besitzer der Versionen 1.0 und 1.1 können gegen Einsendung der Originalversion und einer Updategebühr von DM 198,- die neue Version 2.01 erwerben.

Hyper Soft
Software Design
Hauptstr. 44
5441 Auderath
Tel.: 02676/1863

Neue Chemie-Software

Von Chemo-Soft ist ein neues Produkt namens Kristallotech auf den Markt gekommenm, das sowohl ein Lehrprogramm als auch eine Bildersammlung von Kristallstrukturen beinhaltet. Die Strukturen liegen als hochauflösende Grafiken vor und können von den meisten Programmen weiterverarbeitet werden. Das Lehrprogramm erklärt ausführlich alle wichtige Kristallstrukturen Der Preis beträgt DM 89,-. Weitere Disketten für Fortgeschrittene sind in Vorbereitung.



Ferner wird eine Spezialversion des Programmes Chemotech angeboten, das um einen Programmpunkt erweitert wurde, der es ermöglicht, Korrosionsgrößen und Abtragungsraten zu berechnen. Berechnet werden außerdem noch flächenbezogene Massenverlustraten, Oberflächen von Körpern mit und ohne Bohrung sowie die verschiedenen Dimensionen und Maße, die in der Technik Anwendung finden. Erforderliche Angaben die Maße der Probeform, die Dichte des Körpers, die Angriffszeit sowie das Anfangs- und Endgewicht. Als Grundlage dient die Euronorm 121-72.

Chemo-Soft
Nadorster Str. 81
2900 Oldenburg
Tel.: 0441/82851

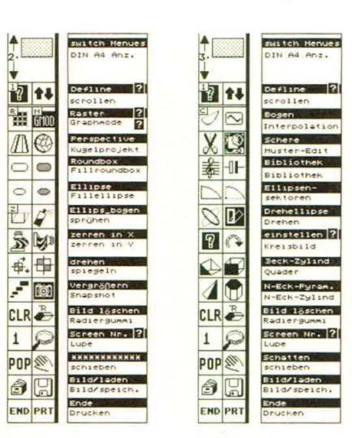
Malprogramm GAMMA zeigt sich im neuen Kleid

Grundlegend aktualisiert wurde das Malprogramm Gamma. In der jetzt erhältlichen Version Gamma_04 finden sich nicht nur neue Funktionen, die Routinen und Algorithmen der älteren Version 03 (siehe Testbericht in ST-Computer 1/89, Seite 64) wurden teilweise auch überarbeitet. Gamma_04 bietet nach wie vor drei Menüleisten, in denen sich teilweise geänderte Funktionen verstekken. Um eine vierte Leiste zu vermeiden, hat der Programmierer nun eine Pop-Up-Box geschaffen, die der "Zeichner" jederzeit durch Drücken der rechten Maustaste erreichen kann. Das Programm bietet unter anderem verschiedene Diagrammdarstellungen an. Mühelos kann der Benutzer Linien-, Balken- und Kuchendiagramme beliebigen Ausmaßes und wählbarer Größe erstellen. In der neuen Box finden sich zusätzlich die Symbole "A3" und "A4". Hiermit kann man sich eine Übersicht über zwei oder vier Bildschirme verschaffen, weil der Bildinhalt verkleinert dargestellt werden kann.

Hinzugekommen sind in der Version 04 auch Rauten, Sechs- und n-Ecke, rechtwinklige Dreiecke, Parallelogramme sowie gefüllte Ellipsensegmente und Segmentlinien. Bei bestimmten geometrischen Figuren gibt das Programm eine Hilfslinie an, die senkrecht auf der im Moment gezeichneten Linie steht. Damit lassen sich zum Beispiel Dreiecke besser zueinander ausrichten. Auch die UNDO- ("ungeschehen machen")-Funktion wurde in die Pop-Up-Box übernommen. Gamma bietet jetzt neben einem Kreis durch drei Punkte auch Spiralen und Füllen mit beliebig großen Blöcken als Füllmuster. Ferner stehen dem "Künstler" Kugel- und Parallelprojektion zur Verfügung. Außerdem kann Gamma_04 auch bemaßen.

Das größte Manko der alten Version - die langsame Drukkausgabe (sie dauerte mitunter sieben bis acht Minuten) - wurde ebenfalls beseitigt. Nun kann man seinen Ausdruck schon innerhalb kürzester Zeit (bei einem NEC P6 plus nach etwa 30 Sekunden) in Händen halten. Der Ausdruck auf 24-Nadeldruckern geht jetzt problemlos vonstatten.

Im Gegensatz zu Gamma_03 kann zwischen allen Bildschirmen zeilenweise hin- und hergescrollt werden. Die ersten beiden Bildschirme werden nicht mehr zu einer DIN A4-Seite zusammengefaßt, sondern der Programmbesitzer kann alle Seiten zusammen bearbeiten. Bezüglich des Ladens und Speicherns von Grafiken kann man wählen zwischen Screenformat (*.PIC, 32000 Bytes), Degas-Bildschirmen (*.PI?, 32066 Bytes), GFA-Objekten (*.OBJ) und DIN A4-Seiten (*.DIN, 64000 Bytes). Das Zusammenspiel von Programm und Festplattenlaufwerk ist bei Version 04 mühelos möglich, da sich das Programm die einzelnen Pfade merkt. GEM-Image-Format (*.IMG, gepackt) und STAD-gepackt (*.PAC) sind nach Angaben der Vertriebsfirma in Vorbereitung.



Die neue Version von Gamma, die wieder auf zwei Disketten geliefert wird, kostet 169,- DM. Das Upgrade jeder beliebigen Gamma-Version beläuft sich auf 50,- DM. Der Softwarehersteller teilte mit, daß das Programm möglicherweise demnächst auch für PCs zu haben sein wird, da gerade an einer Umsetzung in Turbo Pascal gearbeitet wird.

Die Adresse der Vertriebsfirma hat sich geändert und lautet nun:

ASTROSOFT
Hauptstraße 21
6116 Eppertshausen

Griechische Zeichen

Eine original griechische Schreibmaschinentastaturbelegung für den ST bietet das Programm G_SCRIPT ST. Hierbei kann man jederzeit mittels Tastendruck vom normalen lateinischen auf griechischen Zeichensatz und zurück umschalten. Dabei ist man nicht auf eine Textverarbeitung angewiesen, sondern kann dies jederzeit vornehmen (z.B. ein griechisches Wörterbuch mit einer Datenbank anlegen). Da das Programm besonders für Übersetzer interessant ist, lassen sich auch die europäischen Sonderzeichen direkt über die Tastatur eingeben. Ein Druckertreiber für Epson-kompatible und NEC P6-Drucker wird ebenfalls mitgeliefert. Der Preis beträgt DM 249,- inkl. Versand, Handbuch, 5 Zeichensätzen, Zeichensatzeditor und Konverter. Eine Version für DM 149,- mit nur einem Zeichensatz ohne Editor und Konverter ist ebenfalls erhältlich. Eine Aufrüstung auf die Standardversion kostet DM 119,-. Updates können gegen eine geringe Updategebühr bezogen werden.

CTS Haustein
Teutoburger Str. 93
4200 Oberhausen
Tel.: 0208/63687

Neues von BELA

TURBO ST
Neue Version 1.6

BELA bringt in diesen Tagen eine neue Version des Softwareblitters auf den Markt. Die Neuerungen sind unter anderem eine weitere Beschleunigung sowie die Unterstützung von nachladbaren Fonts mit einer maximalen Größe von 8x16 Pixel. Außerdem wurde Problemen mit Programmen, welche GDOS benutzen, beseitigt und die automatische Erkennung des Blitters eingebaut. Registrierte Besitzer von Turbo ST werden über das Update und die Updatemodalitäten automatisch benachrichtigt.

MULTIDESK

Jedem Besitzer eines ATARI ST ist die Beschränkung auf maximal sechs Accessories im Desktop schon unangenehm aufgefallen. Hier bietet BELA mit Multidesk die Hilfe für den ST-Anwender an. Mit Multidesk können beliebig viele Accessories jederzeit vom Desktop aus und in auch in Programmen nachgeladen werden. Jedes Accessories kann aber auch wie ein normales Programm gestartet werden. Pro Multidesk-Fenster können bis zu 32 Accessories installiert werden. Das Besondere dabei ist, daß Multidesk in sich selber installiert werden kann. Multidesk kostet 79,- DM.

G+PLUS
Der GDOS-Ersatz

BELA bietet mit G+Plus einen vollständigen Ersatz für GDOS an. G+Plus ist voll kompatibel zu GDOS und bremst im Gegensatz zu GDOS den Rechner nicht mehr. G+Plus lädt bei Start eines Programms die zugehörige Assign-Datei und die Fonts automatisch und ohne Reset nach. Im Lieferumfang sind der Treiber für den Einsatz im AUTO-Ordner und ein Accessory zur Steuerung der Funktionen enthalten. Das Programm kostet 79,- DM.

STOP
Der Datentresor

BELA bietet ebenfalls ein Accessory namens Stop an, das Laufwerke bzw. Partitionen global, d.h. vollständig codiert. Es sind maximal 16 Passwörter mit je 64 Stellen erlaubt, die über einen Superuserstatus verwaltet werden. Maximal drei User können gleichzeitig am System arbeiten. Mit Hilfe eines eingebauten Cache wird der Zugriff auf die verschlüsselten Daten ohne merkliche Verzögerung bei hoher Datensicherheit durchgeführt. Durch die abschaltbaren Paßwörtern sind auch Arbeitsunterbrechungen am laufenden Rechner möglich. Obwohl Stop ein Accessory ist, kann auch unter TOS auf codierte Daten zugegriffen werden. Stop kostet 129,- DM.

Ebenfalls neu sind noch Hotwire, eine Shell, welche beliebige Programme vom Desktop aus durch einfachen, frei wählbaren Tastendruck starten kann sowie Midimax, daß für Musiker interessant ist. Midimax kann als Accessory installiert werden und bietet umfangreiche Midikontrollfunktionen, die unter anderem auch in Makros vorprogrammiert werde können. Hotwire kostet 79,- DM, Midimax 129,- DM.

SUMMA

BELA wird in Düsseldorf erstmals die fertige Version des Handwerkspakets Summa vorstellen, die auf der CeBIT in Hannover schon beachtliche Resonanz hervorgerufen hat. Alle interessierten Handwerker sollten sich zur Vereinbarung eines Vorführtermins schon jetzt mit BELA in Verbindung setzen.

Preissenkung des D-50 Sorcerer

Eine Preissenkung ist beim D-50 Sorcerer zu vermelden. Er kostet jetzt nur noch DM 99,-. Mit ihm lassen sich Sounds für D-50/550-Synthesizer edieren, verwalten und kreieren.

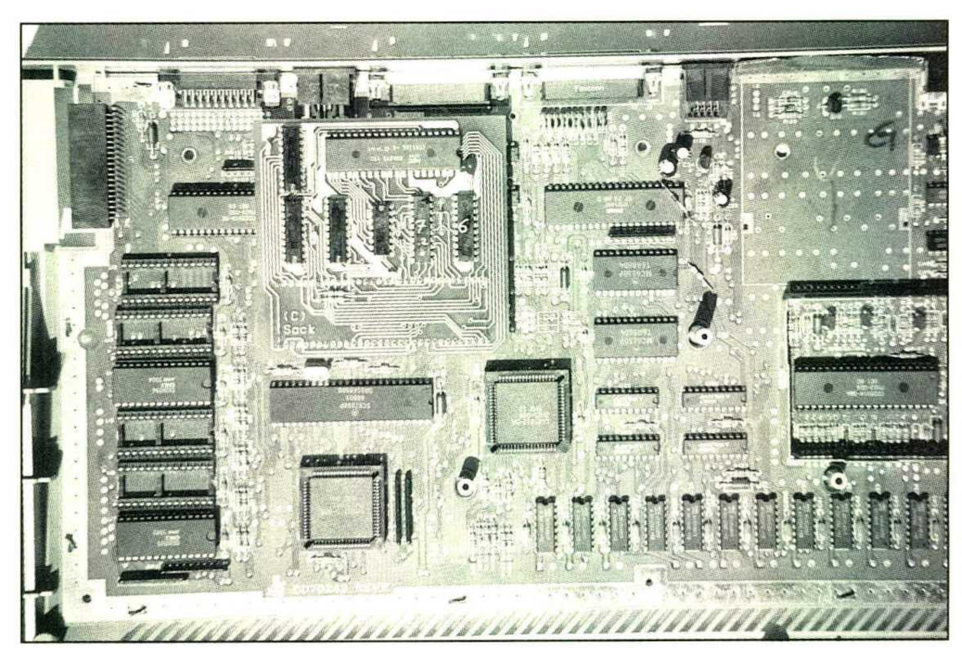
Linsener EDV-Service
Werrastr. 42
1000 Berlin 44
Tel.: 030/6877220


ATARI
DM 5,00
7/89
PD
ATARI PD JOURNAL
PUBLIC DOMAIN JOURNAL FÜR DIE PRAXIS
Kostenlos
Private Kleinanzeigen für Jedermann
Der Katalog
Eine Programmübersicht zum Sammeln
JIL-CAD
PD-Programm
der Spitzenklasse
TOP-TEN
Die besten PD
- monatlich
NEU • Jetzt 7 PD-Serien
mit vielen Neuerscheinungen
NEU • Preise für PD-Autoren
NEU • ALADIN-PD
für den Mac-Emulator
Ab sofort
bei Ihrem Händler oder im Zeitschriftenhandel

PC-Speed

MS-DOS prescht los!

Die Vorliebe für Emulationen steckt vielleicht in der Natur der ST-Anwender oder beruht vielleicht auf Neidgefühlen gegenüber den Bergen an Software, die es für jedes denkbare Anwendungsgebiet in der MS-DOS-Welt gibt, oder entspringt dem Wunsch, aus einem geschlossenen System eine Multimaschine zu machen oder... Was im Endeffekt zutrifft, werden wir mit Sicherheit nie herausbekommen. Tatsache ist, daß der ATARI ST eine ganze Reihe Rechner emulieren kann, u.a. auch einen DOS-Computer. Der Neueste dieser MS-DOS-Emulatoren heißt PC-Speed und ist im Gegensatz zu dem bekannten PC ditto eine Kombination aus Soft- und Hardware.



PC-Speed, der neue MS-DOS-Emulator in einem 520 ST

Eine (un)endliche Geschichte?

Es war einmal vor langer, langer Zeit, als der ATARI ST noch jung war, seine Besitzer ihn noch kaum kannten und von seiner Software nicht recht überzeugt waren. Damals gab es eine Firma (deren Name schon in Vergessenheit geraten ist), die eine graue Box vorstellte, die sich MS-DOS-Emulator für den ATARI ST nannte. Viele Programme in Turbo Pascal konnte man mittels dieser Box auf dem ST bewundern, solange die neugierigen Finger der Anwender nicht wild auf der Tastatur herumfummelten. Abstürze waren seine beliebtesten Kunststückchen, und aus diesem Tiefschlaf konnte man ihn nur mittels der Resettaste zurückholen. Man weiß nicht genau, ob die Einwirkung bösartiger Dämone oder der Einfluß von geheimnisvollen Erdstrahlen die Ursache war, daß die graue Box nicht weiterentwickelt wurde. In jedem Fall irgendwann verschwand sie, und wenn diese Sätze nicht geschrieben worden wären, hätte sich keiner mehr an diese Geschichte erinnert.

Aber die Geschichte - wie ein deutscher Denker einmal sagte - wiederholt sich zweimal: Einmal als Tragödie und das nächste Mal als Kabarettstück. Ebenso verhielt es sich bei den MS-DOS-Emulatoren für den ATARI ST. Nach der Tragödie, jetzt das Kabarettstück: Es gab eine zweite Firma, die auch eine graue Box entwickelt hat (hieß sie nicht Superbox oder so ähnlich?), die eine MS-DOS-Emulation beinhalten sollte. Sie wurde ebenfalls vorgeführt, es wurden Bestellungen entgegengenommen, aber genauso wie ihre Vorgängerin wurde sie nie in Stückzahlen produziert und ausgeliefert.

Zwischen den beiden Emulatorboxen kam ein anderes Produkt auf den Markt, das nicht auf Hardware basierte sondern ein einfaches Programm war, das einen MS-DOS-Rechner tatsächlich emulierte. PC ditto wurde so der einzige funktionierende Emulator für den ATARI ST. Er ist zwar langsam, verleiht dem ST-Anwender aber einen Hauch eines MS-DOS-Rechners.

Aber diese Erzählung wäre keine wahre Geschichte, wenn wir nicht von einer neuen Emulation reden könnten. PC ditto hat eine harte Konkurrenz bekommen. Aus Deutschland kommt ein Hardware-Emulator, der alles verspricht, was schon andere versprochen haben aber nicht zustande brachten. PC-Speed ist die Hauptfigur unserer folgenden Geschichte, die manchen ST-Anwender auftatmen läßt.

Ein Mann sieht blau

Die Gechichte fängt 'tief in Deutschland' an, wo die Sonne nicht ganz verstaubt ist, aber die Luft immer noch nach viel Industrie riecht, die dort ansässig ist. Es ist die Geschichte eines jungen Mannes, den gestört hat, daß viele Anbieter einen Emulator versprochen haben, der zumindest annähernd die Geschwindigkeit eines normalen PC-XT bringen sollte und auf ein wenig Kompatibilität mit einem PC verweisen kann (wohl bemerkt, nicht jeder PC-Clone ist hundertprozentig kompatibel), und dann doch nichts daraus wurde. Sagen wir, er hat solange gewartet, bis ihm klar war, daß die anderen Emulatoren vielleicht nicht mehr kommen würden.

Dann hat er sich an die Arbeit gemacht und hat es geschafft. Auf einer Platine von 9,5x9,5 cm steckt die gesamte Arbeit dieses jungen Mannes. Ein NEC V30-Prozessor (eine Intel-ähnliche CPU), ein Paar PAL-Bausteine und dazu noch 4 TTL-ICs bilden das Herz dieses Emulators. Dazu kommt ein wenig Software, und sie haben auf dem ST einen funktionierenden MS-DOS-Rechner, und was für einen. Es hört sich alles ziemlich einfach an, und Sie werden sich auch fragen, warum nicht schon früher der ein oder andere Hersteller auf die gleiche Idee gekommen ist. Wie gesagt, es hört sich sehr einfach an, aber bis alles so läuft wie es sein soll, muß man hart arbeiten.

Die meisten Versuche der anderen Hersteller sind von einem anderen Prinzip ausgegangen. Sie haben einen Chip entwickelt (oder zumindest teilweise entwickelt), der in der Regel nichts anderes beinhaltete als einen Umsetzer von Intel- in Motorola-Maschinencode. Die Daten wurden von der Software aufbereitet und zum Customchip geschickt. Dieser hat sie dann mehr oder weniger dekodiert und für den 68000-Prozessor verständlich gemacht. Die Speicherverwaltung sowie die Ansteuerung der Peripherie wurde völlig vom 68000er gemeistert. Dieser Transfer von Daten zwischen Hauptprozessor und Customchip, der immer in einem genauen Timing laufen muß, ist nicht einfach zu realisieren und kostet viel Zeit. Aber mit großer Wahrscheinlichkeit muß man die Ursache dieses gescheiterten Versuchs in dem selbstentwickelten Chip suchen, der angeblich nie ganz fehlerfrei war und deswegen nie richtig funktioniert hat.

Bei PC-Speed ist die Lösung irgendwie logisch. MS-DOS-Rechner verstehen nämlich Intel- und keinen Motorola-68000-Code, und deswegen ist es sinnvoller, die Dekodierung dieser Befehle an eine Intel-CPU oder einen kompatiblen Prozessor (V20 oder V30) zu übergeben. Man schafft also im ST einen bestimmten Speicherbereich, der ähnlich wie in einem MS-DOS-Rechner aufgebaut ist. Man setzt den Programmzähler des V30 (in diesem Fall) an die richtige Adresse und läßt diesen so arbeiten als hätte er es mit einem ganz normalen MS-DOS-Rechner zu tun. Die ganze Zeit wird der 68000er in einer Art Tiefschlaf gehalten, damit es nicht zu Konflikten mit dem anderen Prozessor kommt. Wird auf irgendeinen Ein-/Ausgabebereich zugegriffen, werden die Daten aufbereitet und dem 68000er wieder das Kommando übergeben (und das auch nicht immer, es gibt sogar Ausgaben wie z.B. die Videoverwaltung, die der V30 übernimmt). Ist das erledigt, übernimmt wieder der V30 das Kommando usw. Einfach, aber genial.

V30, ein getarnter 8086?

Der V30 ist die Antwort der Firma NEC an den großen Bruder Intel. Mit dieser Prozessorserie hat NEC eine Reihe von Prozessoren geschaffen, die nicht nur mit den herkömmlichen Intel-Prozessoren der 80er-Reihe pinkompatibel sind, sondern sie sogar teilweise an Leistungsstärke übertreffen.

In unserer Geschichte ist der V30 der Protagonist, und wir werden diesem Chip ein wenig Aufmerksamkeit widmen. Der V30 ist eine 16 Bit-CPU, die denselben Befehlssatz beherscht wie der Intel 8086. Dementsprechend kann man diese in einem normalen PC einfach austauschen. Die interne Architektur ist sehr ähnlich, nur die Bezeichnungen der Befehle und Register heißen bei NEC anders. Nicht nur daß man kompatibel geblieben ist, sondern man hat sich auch die Mühe gegeben und Verbesserungen vorgenommen. So ist es z.B. beim internen Bus der Fall: Der 8086 verfügt nur über einen Bus, der V30 aber über zwei. Ein Zähler, der bei dem V30 vorhanden ist, übernimmt die Aufgabe des Schleifenzählers bei Operationen wie Rotieren oder Verschieben. Diese Erweiterung hat zufolge, das z.B. Multiplikations- und Divisionsbefehle von einem V30 wesentlich schneller ausgeführt werden als bei einem 8086.

Neben den alten Befehlen, die auch im Intel 8086 vorhanden sind, wurde der gesamte Befehlssatz mit eigenen Befehlen erweitert, die für bestimmte Anwendungen sehr interessant sein können. Weiterhin muß man erwähnen, daß der V30 einen anderen Prozessor "emulieren" kann, nämlich den 8080. Die Umschaltung erfolgt über das Flag-Register.

Geschwindigkeitstest

PC-Speed wurde von uns zwei Geschwindigkeitstests unterworfen: Der Norton-Utility und dem Performance-Analyzer-Test. Hier sind die Ergebnisse:

```
µDESIGNS    IBM COMPATIBLE PERFORMANCE ANALYZER    (C) 1986
          Copyright (C) 1986 Richard B. Johnson

Checking memory block write    ticks: 68 compared to IBM/PC 742%
Checking register to memory    ticks: 216 compared to IBM/PC 227%
Checking memory to register    ticks: 278 compared to IBM/PC 176%
Checking register to register ticks: 248 compared to IBM/PC 222%
Checking divide by register    ticks: 88 compared to IBM/PC 581%
Checking divide by memory      ticks: 143 compared to IBM/PC 422%
Checking multiply by register ticks: 113 compared to IBM/PC 425%
Checking multiply by memory    ticks: 141 compared to IBM/PC 395%
Checking stack operations      ticks: 256 compared to IBM/PC 173%
Checking far jumps, far calls ticks: 289 compared to IBM/PC 179%

Total time is: 1840 clock ticks, (102 seconds) compared to IBM/PC 280%
```

Zum Vergleich haben wir drei andere Rechner herangezogen, so daß sich folgende Tabelle ergibt:

	PC-Speed V30/8MHz	PC-XT 8088/4,77 MHz	PC-XT V20/4,77 MHz	PC-AT 80286/8MHz/32 Bit-CPU
Norton-Faktor	4,0	1	1,8	9,0
Performance-Test	280%	100%	150%	497%

Wir haben versucht, alles schematisch zu erklären. Natürlich gehört mehr dazu, um eine solche Schaltung exakt im Timing zu realisieren. Aber das sollte die meisten unserer Leser nicht so genau interessieren.

Lötzinn & ein wenig Geduld

Wir kommen jetzt an einen der wichtigsten Teile des Berichts. Wo wird der Emulator im ATARI ST untergebracht? Wir können Ihnen von vornherein den Schreck nehmen: PC-Speed belegt keinen Port (also Sie können auch nebenbei Ihren Aladin in den ROM-Port stecken oder irgendwelche andere Erweiterungen, die Sie ständig benutzen), und man braucht keine Ausbildung zum Elektroniker abgeschlossen zu haben, um den Emulator und den ATARI ST miteinander zu verbinden.

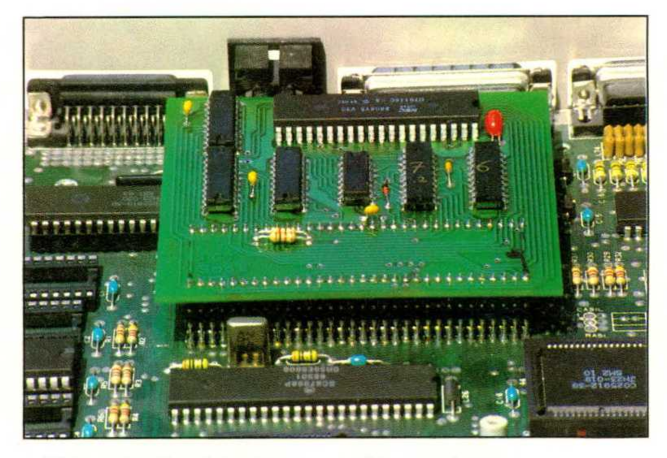
Die Platine besitzt auf ihrer Unterseite einen doppelreihigen Stecker (64poligen IC-Sockel), der direkt auf den Prozessor (68000) gelötet wird. Da alle Bus-Signale und ebenso die Taktfrequenz an diesem Chip anliegen, holt sich der Emulator von dort alles, was er braucht. Auch die Spannungsversorgung holt sich der Emulator von dort. (Dadurch daß der V30 in CMOS-Technologie hergestellt wird, ist der Stromverbrauch sehr gering. Der Prozessor wie der gesamte Emulator werden kaum warm.)

Ein wenig komplizierter wird es beim 1040 ST. Wegen einer Änderung auf der Platine (Motherboard) befindet sich der Prozessor direkt unter der Tastatur, so daß der Platz in der Höhe nicht mehr reicht. Hier hat man zwei Alternativen: Entweder man verzichtet auf den Blechdeckel (der Rechner ist dann nicht mehr abgeschirmt und kann, aber muß nicht, Störungen in Radios u.ä. Geräten verursachen), oder man schneidet einen Ausschnitt aus dem Blech heraus. Aber das ist mit Sicherheit kein größeres Problem. In meinem alten 520 ST fehlte jahrelang das Abschirmblech, da ich ständig an dem Rechner gebastelt habe. Ich kann mich nicht erinnern, daß ich irgendeine Störung im Monitor oder meinem Farbfernseher hatte, der neben meinem Rechner stand. Falls Sie sich trotzdem nicht trauen den ST zu öffnen und PC-Speed selbst zu montieren, dann fragen Sie Ihren Händler oder der MAXON Computer nach.

Ist alles sauber gelötet (bitte überprüfen Sie Ihre Lötstellen, bevor Sie den Rechner erneut anschalten), dann können Sie eine Diskette einlegen und den ersten Versuch starten. An der Software wurde nicht gespart und auf der mitgelieferten Diskette befindet sich alles, was Sie brauchen, um den Emulator in Gang zu bringen. Ein DOS-Betriebssystem ist allerdings nicht vorhanden. Das müssen Sie sich selbst besorgen.

Das erste Programm, das Sie starten sollten, ist das Initalisierungsprogramm. Mit ihm können Sie Ihre Rechnerkonfiguration festlegen. Also ob Sie eine CGA- oder eine Hercules-Grafikkarte emulieren möchten, ob Sie ein oder mehrere Laufwerke oder ein 5 1/4"-Laufwerk besitzen, ob Sie eine Festplatte betreiben, wieviele Partitionen Sie dann einrichten möchten, von welcher Partition Sie booten möchten und, und, und...

Das zweite Programm ist die Emulator-Software selbst. Wenn sie PC-Speed starten, werden Sie aufgefordert, ein DOS-System von Ihrem internen Laufwerk oder von Ihrer Boot-Partition zu laden, falls Sie eine solche haben. Ist das alles geschehen, sitzen Sie immer noch vor Ihrem ST, aber intern läuft ein MS-DOS-Rechner. Das Programm ist resetfest, und Sie können jederzeit durch Drükken der ESC-Taste den Emulator verlassen. Falls Sie sich allerdings gerade in einem DOS-Programm befinden, ist das Verlassen gesperrt, da viele DOS-Programme diese Taste häufig nutzen. Noch ein Programm ist im Lieferumfang enthalten. Es dient dazu, die verschiedenen Maustypen, die es für den PC gibt, zu unterstützen.



PC-Speed muß auf den Prozessor gelötet werden.

Tauglichkeit ist (k)ein Gebot

"Bis jetzt läuft noch alles gut!", sagte der Typ, der vom Dach eines Hochhauses stürzte und gerade an der 15. Etage vorbeikam. Was ist mit der Kompatibilität? Was läuft und was nicht, werden Sie sich jetzt fragen? Das ist immer eine sehr schwierige Frage. Ich kenne von verschiedenen Messen und von Anrufen in der Redaktion her, daß jeder gerne wissen möchte, ob das Programm, das er am meisten benutzt, einwandfrei auf diesem Emulator funktioniert. Die Frage ist schwer zu beantworten, da eine unheimliche Menge an Software in der MS-DOS-Welt existiert. Um alles zu testen, müßte man zum einen alle vorhandenen Programme besitzen und zum anderen benötigte man unendlich viel Zeit.

Programme wie WordPerfect, Turbo Pascal, Lotus 1-2-3, Autocad oder Sidekick laufen problemlos. Ob allerdings irgendein "exotisches" Programm, das wir - verständlicherweise - nicht testen konnten, läuft, können wir natürlich nicht sagen. 100% Kompatibilität findet man aber auch bei einem reinen PC-Clone nicht. Man muß hinzufügen, daß viele Programme, die sehr schwer geschützt sind (meistens Spiele), mit großer Wahrscheinlichkeit nicht laufen werden. Aber alles, was wir probieren konnten, läuft einwandfrei und sehr schnell.

Es muß bunt sein & viele Punkte haben

Eine der vielen Fragen, die jeder Käufer einer Emulation sofort stellt, ist: Kann der Emulator Hercules-Auflösung oder so viele Farben wie eine VGA-Karte? Man muß sich darüber klar sein, das ein Emulator die bestehenden Fähigkeiten (oder Unfähigkeiten) des ST nicht erweitern kann. Eine PC-Hercules-Karte verfügt horizontal über eine höhere Auflösung als der ST, nämlich 720 Pixel. Woher soll der ST die fehlenden 80 Bildpunkte nehmen? Auch im Bereich Farbe ist der ST sehr eingeschränkt. Die Farbmöglichkeiten einer VGA-Karte wird der ST ohne zusätzliche Grafikkarte nie erreichen. Also wie Sie sehen, liegen die Einschränkungen dieser Emulation nicht bei der Emulation selbst sondern auf der ATARI ST-Seite.

PC-Speed kann trotz dieser Einschränkungen, die der ATARI ST von vornherein festlegt, eine ganze Menge: CGA- und

Monochromkarte. Eine Hercules-Karte wird nicht ganz dargestellt, aber unterstützt. Das bedeutet, daß Programme, die nur in dieser Auflösung laufen, nicht abstürzen und der Bildschirm mit den Cursor-Tasten nach links und rechts gescrollt werden kann. Also auch in diesem Punkt wurde Optimales geschaffen.

Welche Programme laufen mit PC-Speed?

Wie schon im Artikel erwähnt wird, können wir nicht alle Programme auf PC-Speed laufen lassen. Wir zeigen Ihnen hier eine Liste der Programme, die wir selbst getestet haben:

Microsoft Word	Turbo BASIC
Wordstar	GBASIC
Microsoft Char	dBase III
Sidekick plus	Lotus 1-2-3
Turbo Pascal	Flugsimulator
Turbo C	Golf

Es funktionierten ebenso verschiedene DOS-Versionen:

DOS 2.1
DOS 3.3
DOS 4.0

Auch hier konnten wir nicht alle existierenden DOS-Versionen überprüfen, da zu fast jedem PC ein anderes DOS mitgeliefert wird.

Massenspeicher & Außenwelt?

Wir haben schon oben erwähnt, daß PC-Speed eine Festplatte unterstützt. Aber welche? Es gibt nämlich mittlerweile eine ziemlich lange Liste verschiedener Anbieter, die sehr unterschiedliche Platten für den ST herstellen. PC-Speed kann mit fast allen arbeiten, die ganz normal über den DMA-Port angeschlossen werden und den Festplattentreiber ATARIs unterstützen. Schwierigkeiten gibt es bei Festplatten, die - wie die in der *c't* vorgestellte "Billiglösung" - einen eigenen modifizierten Treiber brauchen (aber sogar in diesem Bereich wird gearbeitet und demnächst wird es auch für diese Platten eine spezielle Lösung geben). Auch hier ist es sehr schwierig, ein Urteil abzugeben und jede Festplatte, die mit PC-Speed läuft, mit Namen zu nennen. Wir haben es mit einer SH205, einer Vortex HDplus 20 und 30 probiert, sie sind einwandfrei gelaufen. Es kann aber wohl möglich sein, daß die ein oder andere Festplatte anderer Hersteller nicht läuft. Das wird aber nur die Zukunft zeigen können. PC-Speed kann natürlich auch ein externes Laufwerk unterstützen, das nicht unbedingt ein 3,5"- sondern ohne weiteres ein 5.25 "-Laufwerk sein kann. Man kann je nach DOS-Version 40 oder 80 Spuren formatieren.

Paralleler und serieller Port sowie die Maus werden ebenfalls von PC-Speed voll unterstützt. Auch die Soundmöglichkeiten werden voll ausgenutzt.

Speedy Gonzales

Ein Phantom geht in der PC-Welt um, das Phantom der Geschwindigkeit. Alle Entwicklungsabteilungen der Chiphersteller bemühen sich, den höchsten Norton-Faktor zu erreichen. Der, der nicht mindestens auf Faktor 1 kommt, kann seine Entwicklung todsicher in die Schublade stecken. Ja, Herr Norton hat sich eine Reihe von Benchmarkoperationen ausgedacht, die einen gewissen Geschwindigkeitsfaktor messen sollen. Wohl bemerkt, es ist nicht nur Herr Norton, der so etwas gemacht hat. Es gab und gibt (und, weiß Gott, wieviele es noch geben wird) viele andere, die sich solche immer 'raffinierteren' Geschwindigkeitstests ausgedacht haben.

Um was geht es bei diesen Tests? Hauptsächlich um extreme, interne Befehlsabwicklungen wie z.B. Integer-Multiplikationen, die nur über Register laufen, ähnliche Divisionen oder Transfer-Befehle usw. Am besten schauen Sie sich mal den Infokasten mit den Ergebnissen an.

Sind diese Tests aussagekräftig? Nur bedingt. Z.B. hat der V30 einen schnelleren Multiplikationsbefehl als ein 8086-Prozessor. Folglich schneidet er bei einem direkten Vergleich auf jeden Fall besser ab. PC ditto, der Software-Emulator, hat bei einer Blockoperation (MOVE - von Adresse zu Adresse) eine doppelt so hohe Wertung wie ein normaler PC-XT, und jeder, der PC ditto gesehen hat, weiß ganz genau, das er langsamer ist als ein PC-XT. Der normale Anwender hat von solchen Vergleichstests nicht viel. Es ist mehr oder weniger eine Prestigefrage, zu sagen, "Mein PC hat ein Norton-Faktor 3,678", oder ähnliches. In jedem Fall ist PC-Speed auch in diesem Bereich ein sehr schneller PC. Mit einem Norton-Faktor 4.0 ist er quasi 4mal schneller als ein PC-XT mit einer Taktfrequenz von 4,77 MHz.

Wichtig ist, daß man mit diesem Emulator problemlos und genau so schnell arbeiten kann wie mit einem normalen PC-AT, und das ist das Entscheidende. Also wenn man das Prestige, das ein Norton-Faktor verleiht, braucht, hat PC-Speed sehr gut abgeschnitten, und wenn es praktisch darum geht, tatsächlich schnell zu sein, trifft das ebenso zu. Man bekommt für DM 498,- einen sehr schnellen MS-DOS-Rechner.

PC-Speed stellt genau wie PC ditto maximal 704 kByte freien Speicher zur Verfügung. Er läuft auf allen ST-Modellen, sogar auf einem 520 ST mit 512 kByte Speicher und Disketten-TOS (es stehen dann ca. 260 kByte für PC-Speed zur Verfügung).

Fazit

Was soll man als Schlußwort sagen, was nicht schon gesagt ist? Viele der über 300.000 ATARI-Besitzer haben sehr lange auf eine DOS-Emulation dieser Art gewartet. Viele davon haben sich PC ditto gekauft, weil er der einzige lauffähige Emulator war, der den ST in eine PC-Welt versetzte. Heute ist es Realität geworden, daß ein Hardware-Emulator da ist. MAXON Computer hat den Entwickler zu sich auf den Stand der ATARI-Messe in Düsseldorf eingeladen. Sehen Sie selbst und überzeugen Sie sich!

MM

Bezugsadresse:

Hans Sack
Bleichstr. 49
4792 Bad Lippspringe
Tel.: 05252/4290

Probleme und Fragen?

Da die MAXON Computer bereits mehrere Jahre Erfahrungen beim Vertrieb des PC dittos gesammelt hat und auch PC-Speed anbieten wird, wissen wir sehr gut über die Probleme mit Emulatoren Bescheid. Falls Sie noch Probleme und Fragen zu PC-Speed haben, können Sie uns telefonisch unter 06196/481811 erreichen.

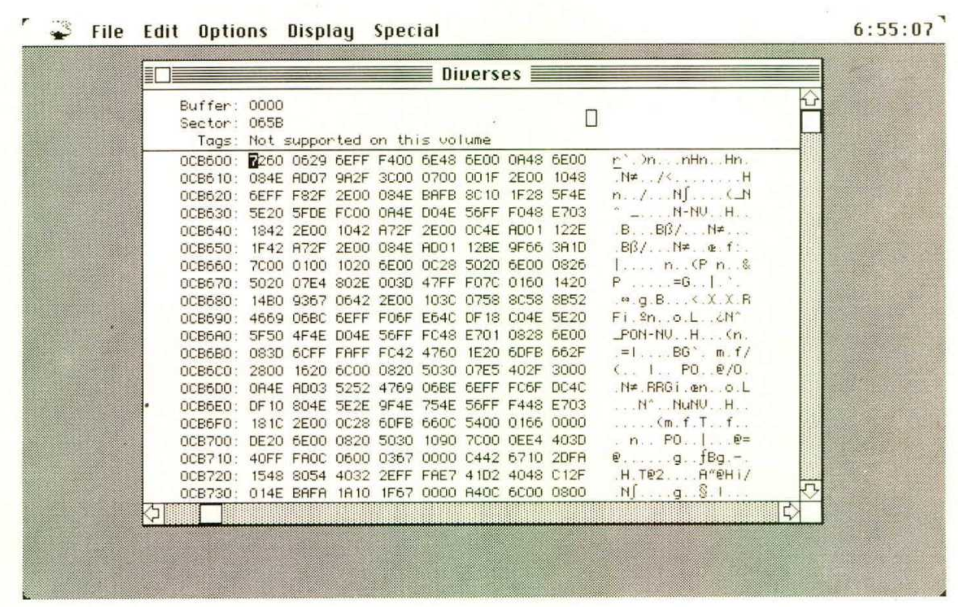
Scheibenkleister à la Aladin

Das Aladin-Diskettenformat

Ok! Da lag sie also vor mir, die frisch formatierte Aladin-Diskette. So so! Hm? Sieht eigentlich genauso aus wie eine GEM-Diskette! Und was ist da nun drauf? Hat mich wirklich interessiert. Nach mehreren anfänglichen Fehlversuchen, (Schere, Pinzette, Lupe, Mikroskop,...), dem Wahnsinn nahe, gab mir ein Freund den heißen Tip, es einmal mit einen Diskeditor zu versuchen. Einigermaßen erholt, hier nun der Bericht von der Reise durch Aladins Bits und Bytes.

Wer schon einmal eine Aladin-Diskette unter GEM eingelegt hat, dem ist aufgefallen, daß Aladin-Disketten ein richtiges GEM-Directory besitzen. Es ist jedoch "nur" ein Mini-GEM-Directory, das einen gewaltigen Absturz unter GEM beim Einlegen der Diskette verhindert. Der würde nämlich auftreten, wenn es nicht vorhanden wäre. Andererseits kennzeichnet es die jeweilige Diskette auch unter GEM als Aladin-Diskette. Ziemlich nützlich, wenn man mal wieder vergessen hat, eine Diskette zu beschriften. *"Was ist denn auf dieser verdxxxxxx Diskette schon wieder drauf ?"*

Sogar im GEM-Bootblock befindet sich ein kleines Programm, das beim Versuch eine Aladin-Diskette unter GEM zu Booten eine Meldung auf dem Bildschirm ausgibt. Anscheinend waren sich die Aladin-Entwickler wohl bewußt, wie es auf manchen Computerschreibtischen bzw. in manchen Diskettensammlungen zugeht.



Komfortables Edieren von Aladin-Dateien ist mit dem PD-Diskeditor FEdit möglich.

En Block

Alle Speichermedien unter Aladin arbeiten blockorientiert. Aladin kennt weder Tracks noch Sektoren, die er verwalten muß. Ein Speichermedium beginnt bei Block 0 und hat eine bestimmte Größe, sprich Anzahl Blöcke. Eingefleischte Aladianer sprechen von *Allocation Blocks*. Alle Speichermedien werden gleich verwaltet. Aladin macht keinen Unterschied ob RAM Disk, Diskette oder Harddisk. Die Größe der Blöcke kann von Speichermedium zu Speichermedium unterschiedlich sein. Bei MFS-Formaten steht nur eine begrenzte Anzahl von Blöcken zur Verfügung. Daher wächst mit steigender Diskettengröße auch die Größe der Blocks. Die Blockgröße schwankt somit je nach Größe des Mediums zwischen 512 Byte (360k-Diskette) und mehreren Kilobytes (Partition auf der Harddisk). HFS-Formate können beliebig viele Blöcke und ein beliebig großes Directory haben. Die Blockgröße liegt hier meistens bei 1024 Byte.

Ein kleiner Tip für diejenigen, die sich eine Aladin-Diskette einmal mit einem GEM-Diskeditor anschauen möchten. Bei Disketten beginnt die eigentliche Aladin-Diskette (Block 0) auf Track 1 / Sektor 0. Dies gilt jedoch nur für 360/720k-Disketten. Dazu später mehr.

Boot Blocks Version	22
System File Name	System
Finder File Name	Finder
Debugger File Name	MacsBug
Disassembler File Name	Disassembler
Startup Screen File Name	StartUpScreen
Initial File to Run	Open Power
Clipboard File Name	Clipboard File
Max Number of Open Files	20
Max Number of Events	20
System Heap Size (128K)	17152
System Heap Size (512K)	98304

Update

Cancel

FEdits Dialogbox für die Bootblocks

Die ersten beiden Blocks einer Aladin Diskette, wie könnte es auch anders sein, sind etwas ganz Besonderes.

Die Bootblocks bestimmen über die Bootfähigkeit der Aladin-Diskette, d.h. nur wenn sich eine Systemdatei auf der Diskette befindet und in den Bootblocks auch die Bootfähigkeit markiert ist, läßt sich Aladin mit dieser Diskette booten. Dieser Eintrag wird automatisch vorgenommen, wenn man eine Systemdatei auf eine Diskette kopiert.

In den Bootblocks befinden sich noch eine ganze Reihe anderer wichtige Parameter. So z.B. die aktuellen Namen der wichtigsten Systemdateien und die Größe des *System Heaps*. Der System Heap ist ein reservierter Speicherbereich variabler Größe, der nur dem Betriebssystem zur Verfügung steht. Für Anwendungsprogramme gibt es den *Application Heap*, in dem Programme... ; aber wir wollen ja nicht abschweifen. Ganz komfortabel lassen sich die Bootblocks mit *FEdit* bearbeiten. Der PD-Diskeditor für Aladin stellt hierfür eine eigene Dialogbox zur Verfügung.

Das eigentliche Directory einer Aladin-Diskette beginnt ab Block 2. Ich möchte an dieser Stelle gar nicht weiter auf den exakten Aufbau des Aladin-Directorys eingehen. Das Schaubild zeigt den Aufbau eines MFS-Directorys. Den Findigen sollte das genügen. Wer tiefer in diese Materie einsteigen möchte, dem kann ich nur *Inside Macintosh I-III* wärmstens ans Herz legen.

Unter Aladin gibt es genau wie unter GEM auf jeder Diskette eine Desktopdatei. Bei Aladin ist sie jedoch unsichtbar und wird automatisch mitgeführt. Daher entfällt das von GEM bekannte lästige manuelle *Desktop sichern*. In Aladins Desktopdatei werden automatisch die Icons aller Programme und Dokumente abgelegt, die sich auf der Diskette befinden. Auch die Zuordnung zwischen Dokumenten und Programmen wird hier gespeichert. Der Vorteil: Da Aladin jederzeit weiß, welches Dokument zu welchem Programm gehört, lassen sich alle Dokumente vom Schreibtisch aus mit einem Doppelklick öffnen. Falls es sich um eine MFS-Diskette handelt, wird in der Desktopdatei auch noch die Ordnerstruktur der Diskette gesichert.

Die Desktopdatei kann auch Urheber für Probleme unter Aladin sein. Falls Sie beschädigt wird, gibt es Probleme. Folge: Wenn Aladin anfängt, die Partitionen der Harddisk eine nach der anderen auf dem Bildschirm aufzureihen, bleibt er stehen, bevor er eine ganz bestimmte Partition auf dem Schreibtisch erscheint. Auch ein Reset behebt das Problem nicht. Wahrscheinlich ist die Desktopdatei der entsprechenden Partition ist defekt. Abhilfe schafft hier ein kleiner Trick. Man kann Aladin dazu zwingen, die Desktopdatei neu aufzubauen. Durch Drücken der Tastenkombination *Control-Alternate* unmittelbar nachdem die letzte Partition mit intakter Desktopdatei auf dem Schreibtisch erscheint, wird eine ganz spezielle Dialogbox auf den Bildschirm gebracht. Durch Anklicken von *OK* wird Aladin beauftragt, die Desktopdatei der entsprechenden Partition anzulegen.

Byte Nr.	
0	Flags
1	Versionsnummer
2	Desktop Info
18	Filenummer
22	1. Allocation Block der Data Fork
24	Logisches Dateiende der Data Fork
28	Physikalisches Dateiende der Data Fork
32	1. Allocation Block der Resource Fork
34	Logisches Dateiende der Resource Fork
38	Physikalisches Dateiende der Resource Fork
42	Erstellungsdatum
46	Datum letzte Modifiktion
50	Länge des Dateinamens
51	Dateiname

Aufbau eines Directory Eintrages

MFS & HFS

Unter Aladin gibt es grundsätzlich zwei verschiedene Arten der Dateiverwaltung und des Directoryaufbaus, MFS und HFS. Aladin unterstützt standardmäßig das MFS-Directoryformat. **FS** bedeutet in diesem Fall ***File System***. Ob das **M** für den Nachnamen des Programmierers (Sergio **M**accaroni) oder den Firmennamen eines bekannten Bonbonherstellers steht (Kleiner Tip: es ist nicht Suchard), ist bis heute noch ungeklärt. Die MFS-Dateiverwaltung erlaubt keine echte Baumstruktur. Ordner bzw. Subdirectories werden zwar auf dem Schreibtisch angezeigt, jedoch existiert die Zuordung von Dateien und Ordnern auch nur hier. Alle Dateien liegen auf der gleichen Ebene. Daher dürfen Dateien, auch wenn sie sich in unterschiedlichen Ordnern befinden nicht den gleichen Namen besitzen. Für Anwendungsprogramme ist die ganze Ordnerstruktur von MFS einfach nicht vorhanden.

HFS steht für ***Hierarchisches File System***. Was sagen soll, ein Dateiverwaltungssystem mit Baumstruktur und echten Subdirectories. HFS erlaubt eine schnellere und effizentere Verwaltung von großen Speichermedien als MFS. Wenn Sie Aladin mit Harddisk einsetzen, empfiehlt sich HFS auf jeden Fall. Sollte sich die Datei *Hard Disk 20* auf Ihrer Aladin-Systemdiskette befinden, so wird diese und damit auch die HFS-Fähigkeit beim Booten nachgeladen. Aladin quittiert die Anwesenheit der Datei mit der Meldung *HFS installiert*. Von nun an stehen dem Aladin-Benutzer die Segnungen der neuen Dateiverwaltung fast uneingeschränkt zur Verfügung. Lediglich die Systemdiskette bzw. Systempartition muß weiterhin MFS bleiben. Denn HFS-Disketten können unter Aladin nicht gebootet werden. Ist HFS einmal installiert, bekommen ab jetzt alle doppelseitigen Disketten und Festplattenpartitionen beim Formatieren HFS-Directoryformat.

Mehr als nur ein Buch! Mehr als nur Software!

Scheibenkleister II

von innen: Teil II

Aus gegebenem Anlaß: HYPERFORMAT definiert wieder einmal die Grenzen der ST-Disketten neu. War es vor zwei Jahren das erste Programm, das 11 Sektoren auf eine Spur packte, so legen wir heuer erst recht nach: Mit einem minimalen Hardware-Trick passen nun 14 Sektoren auf eine Spur und sorgen mit maximal 1.2 Megabytes für genügend Backup-Reserven. Doch nicht nur damit glänzt die aktuelle HYPERFORMAT-Version, die dem SCHEIBENKLEISTER II beiliegt.



Das Konzept dieses Formatierprogramms: Alles erlauben. Viele frei einstellbare Parameter ermöglichen Ihnen Ihr eigenes Spezialformat für die unterschiedlichsten Zwecke. Und für die weniger experimentierfreudigen Anwender liegen die wichtigsten Standardformate (720 kB, 820 kB, 900 kB, 1.1 MB, 1.2 MB) auf Funktionstasten.

Und was bietet dieses Chamäleon nicht alles:

- MS-DOS-kompatible Formate (auch 40spurige)
- Sektoranzahl einstellbar von 1 bis 14 Sektoren pro Spur
- Schnelladedisketten möglich durch Spiralisierung, Interleave-Tricks und zusätzliche Sektorvorspänne
- Von Schnelladedisketten wird ohne jede zusätzliche Software maxi mal doppelt so schnell geladen!
- Steprate und Kopfberuhigungszeit einstellbar, um jedes Laufwerk gerecht zu werden
- Fixe Formatierroutinen: 20 Sekunden für eine einseitige Disketten 35 Sekunden für eine doppelseitige
- Zuschaltbares Verify; defekte Sektoren werden beim Formatieren aussortiert - auch teilweise defekte Disketten können also weiterverwendet werden
- "Softformatieren" - Diskette löschen in zwei Sekunden
- Systemdisketten mit ausführbarem Bootsektor erzeugbar
- Clustergröße, Laufwerk, Größe des Wurzelverzeichnisses Spuranzahl, Diskettenname wählbar

Kuno K. ist verzweifelt: Startete er doch gerade sein bisheriges Lieblingsformatierprogramm von seiner Systemdiskette, worauf es ihm dieselbe ohne Rücksicht fein säuberlich löschte... mit HYPERFORMAT stäke heute keine Axt in Kunos Rechner: Es formatiert rückwärts (ein typisch bayrisches Programm?). Herkömmliche Formatierer beginnen bei Spur 0 und löschen so sofort alle Verwaltungsdaten (FATs, Wurzelverzeichnis). HYPERFORMAT beginnt am anderen Ende - so haben Sie noch Zeit, um zu bemerken, daß Sie gerade Ihre Systemdiskette formatieren... Sie sehen: HYPERFORMAT ist für die harte Praxis geschrieben.

Ada Rist flucht leise vor sich hin: Wollte sie doch gerade einen Text aus ihrem Textprogramm abspeichern, nur um festzustellen, daß auf ihrer Textdiskette kein Platz mehr ist. Und eine Formatieroption hat das Textprogramm nicht... HYPERFORMAT funktioniert nun auch als Accessory und kann so aus jedem GEM-Programm aufgerufen werden. Ada seufzt erleichert.

Und das Beste: Den umfangreichen, heftig kommentierten Quelltext mit über 100 kB Umfang finden Sie ebenfalls auf der KLEISTERSCHEIBE, der Buchdiskette zum SCHEIBENKLEISTER - unsere Tricks sind somit ein offenes Geheimnis für alle Lernbegierigen.

Ob man dieses HYPERFORMAT auch einzeln kaufen kann? Bedaure - Sie werden schon den Frosch "SCHEIBENKLEISTER" schlucken müssen, um an das Programmchen zu kommen. Aber schließlich erklärt Ihnen selbiges Machwerk auch, wie das alles funktioniert, wie Sie selbst den Floppycontroller des ST bis zum äußersten Wahnsinn treiben, wie man die ST-Floppies per Hardware-Trick auf bis zu 14 Sektoren pro Spur ausreizt, welche geheimnisvollen Tricks mit HYPERFORMAT noch möglich sind... und was sonst noch im KLEISTER versteckt ist, erfahren Sie in den restlichen siebenhundertzweiunddreißig ungemein spannenden Folgen unserer Serie "SCHEIBENKLEISTER II von innen".

```
===== H Y P E R F O R M A T   3.20 =====
===© 1989 Claus Brod & Jürgen Stessun===
T Last track           : 82
D Directory entries    :112
S Sides                : 2
I Interleave factor    : 6
P Sectors per track    : 11
F Spiral factor        : 2
R Step rate (ms)       : 3
C Cluster size         : 2
V Verify count         : 0
Y Settling delay       : n
E Executable bootsector : n
A Additional header    : n
U Disk unit            : A
Q Quickformat          : 0
=============Special keys:=============
CR Format B Softfmt ESC Quit HELP Help
=============Default keys:=============
F1 ATARI 1S  | F2 ATARI 2S | F3 HYPER 1S
F4 HYPER 2S  | F5 FAT 1S   | F6 FAT 2S
F7 MSDOS 2S  | F8 1.1 MB   | F9 1.2 MB
```

Wird Ada einen billigen Zehnerpack Disketten ersteigern können, um sich von ihren Platzproblemen zu befreien? Kann Kuno die sterblichen Reste seines Rechners an einen mysteriösen Bastler mit russischem Akzent verkaufen? Schalten Sie auch nächstes Mal auf Empfang, wenn Sie Doktor Kleister sagen hören:

SCHEIBENKLEISTER II - Massenspeicher am ST.
Mehr als 1.2 MB Software,
870 Seiten Buch.
Für 79 DM.



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ____________________
Vorname: ____________________
Straße: ____________________
Ort: ____________________
Unterschrift: ____________________

Hiermit bestelle ich:

☐ Exemplare von "Scheibenkleister II, Massenspeicher am ST" mit Diskette für DM 79,00

☐ Vorauskasse
☐ Nachnahme

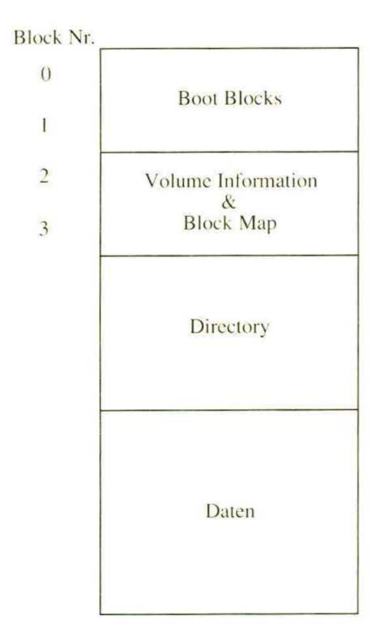
Versandkosten: Inland DM 7,50
Ausland DM 10,00
Auslandsbestellungen **nur** gegen Vorauskasse
Nachnahme zuzgl. DM 4,00 Nachnahmegebühr.

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrieb in Frankreich: AROBACE 2Rue Piemontesis F-75018 Paris Tel: 1/42235044 Fax 1/42545631

Von Janus & anderen Formaten

Apropos Formatieren: Disketten werden unter Aladin mit der Option *Diskette löschen* auf dem Schreibtisch oder jederzeit einfach durch Einlegen einer Leer- oder GEM-Diskette formatiert. In beiden Fällen erscheint die Formatier-Dialogbox.

Normalerweise werden Disketten unter Aladin entweder einseitig mit 360 oder doppelseitig mit 720k formatiert. Die physikalische Formatierung entspricht dabei exakt der von GEM, d.h. 9 Sektoren pro Track und 80 Track pro Seite. Beim Formatieren wird automatisch auch der GEM-Bootblock und das Mini-GEM-Directory auf die Diskette gebannt. Um die Zugriffsgeschwindigkeit zu erhöhen, wird auf jeden Track noch der Header eines 10ten Sektors geschrieben. Dadurch wird die Zeit, die der Disk Controller benötigt, um von einer Spur auf die nächste zu fahren und zu prüfen, ob er auch den richtigen Track erwischt hat, erheblich verkürzt. Dadurch und durch den 1:1 Interleave-Faktor den Aladin zur Formatierung verwendet, ist es möglich Disketten mit der maximalen Geschwindigkeit von 22kByte / Sekunde zu lesen und zu schreiben. Schneller geht´s nun wirklich nicht mehr. Die 360/720k-Disketten lassen sich problemlos mit jedem GEM-Diskcopy-Programm kopieren. Wird dazu das PD-Programm *FCopy* benutzt, bleibt sogar das *Fast Load Format* erhalten.



Aufbau des Volumes

Was nur Eingeweihte wissen oder solche die sich die Mühe gemacht haben, die Aladin-Technotes zu lesen: Mit Aladin lassen sich noch ganz andere Diskettenformate erzeugen. Drückt der findige Benutzer nämlich die *ESC*-Taste während die Formatier-Dialogbox auf dem Bildschirm zu sehen ist, werden die angebotenen Formate um eine Vielzahl erweitert.

Mit den 400/800k-Optionen lassen sich Disketten erzeugen, die exakt die Kapazität einer Original "Tomaten"... Pardon: "Birnen"-Diskette besitzen. Die Disketten werden ein- bzw. doppelseitig mit 10 Sektoren pro Track und 80 Tracks pro Seite formatiert. Verständlicherweise entfiel das GEM-Directory aus Platzgründen . So lassen sich bequem Disketten mit höherer Kapazität erzeugen. Auch bei diesem Format bediente man sich eines kleinen Tricks, um die Zugriffsgeschwindigkeit zu erhöhen. Für einen weiteren Header wie bei den 360/720k-Disketten war allerdings kein Platz mehr. Daher nimmt Aladin nicht den Indeximpuls des Laufwerks als Startsignal zum Formatieren eines Tracks sondern verschiebt die einzelnen Tracks um jeweils 2 Sektoren gegeneinander. Auch hier wieder, um dem Disk Controller genügend Zeit zu geben und doch nach der kürzt möglichsten Zeit auf dem nächsten Track weiterlesen zu können. Besonders wichtig sind 400/800k-Aladin-Disketten jedoch, da sie sich mit Hilfe des Eickmann EXChangers problemlos in "Tomaten"-kompatibles Format umwandeln lassen. Auch 400/800k-Disketten lassen sich mühelos mit einem GEM-Kopierprogramm duplizieren (z.B. mit *FCopy*, Parameter: 10 Sektoren, Disk Def Info = No, ein- oder doppelseitig).

Und noch ein raffiniertes Diskettenformat verbirgt sich in Aladin, Janus. Janus-Disketten sind doppelseitige Disketten, deren Vorderseite eine einseitige ST-Diskette und deren Rückseite eine einseitige Aladin-Diskette beinhalten. Was für einen Sinn das macht? Für die Entwickler von Aladin bestimmt. Die bei Aladin mitgelieferte "ST"-Diskette ist eine solche Janus-Diskette. Auf der Vorderseite GEM, d.h. Aladin mit allen benötigten Dateien. Ist Aladin jedoch gestartet, der Schreibtisch auf dem Bildschirm, legt man dieselbe Diskette noch einmal ein. Es erscheint eine einseitig Aladin-Diskette auf dem Desktop. Voll mit Technotes, Inits und Utilities für Aladin.

Um Janus-Disketten zu erstellen, müssen diese erst unter GEM einseitig formatiert werden. Danach die Diskette unter Aladin noch einmal mit der Janus Option formatieren.

Handwerkszeug

Und womit bin ich nun zwischen Aladins Disketten herumgestromert? Die größte Hilfe war *FEdit*. Der PD-Diskeditor für Aladin ist sehr komfortabel und einfach zu bedienen. Festplattenpartitionen, Diskettenlaufwerke und Aladins Superdisk können gleichermaßen unter die Lupe genommen werden.
Doch Vorsicht: Nie mit einem Diskeditor arbeiten, wenn man nicht wirklich weis, was Mann oder Frau tut. Anfängern sei empfohlen, sich an der Kopie einer unwichtigen Diskette zu versuchen und die

Byte Nr.	
0	$D2D7
2	Datum der Initialisierung
6	Datum des letzten Backup
10	Volume Attribute
12	Anzahl der Dateien
14	1. Block des Directotries
16	Länge des Directotries
18	Anzahl der Allocation Blocks
20	Größe der Allocation Blocks
24	Anzahl der zubelegenden Bytes
28	1. Allocation Block in der Block Map
30	nächste freie File ID
34	Anzahl der freien Allocation Blocks
36	Länge des Volumenamens
37	Volumename

Aufbau der Volume Information

Festplatte erst einmal außer acht zu lassen.

Und noch ein paar Tips von einem alten Aladin-Hasen: Defekte Aladin-Disketten, gelöschte Dateien oder Festplatten lassen sich hervorragend mit den *Symmantec Utilities for the Macintosh* kurz auch *SUM* genannt wiederherstellen. Die SUMs arbeiten einwandfrei unter Aladin und haben mir schon so manchen Dienst erwiesen.

Die Performance der Harddisk läßt nach einigem Gebrauch unter Aladin nach. Dateien werden fragmentiert, d.h. in mehreren Teilen auf der Platte angelegt. Auch die Desktopdatei wächst und wächst mit der Zeit. Jedesmal, wenn eine neue Datei auf die Festplatte kopiert wird,

werden eine ganze Menge von Informationen in die Desktopdatei aufgenommen. Beim Löschen der Dateien wird zwar der Platz auf der Platte wieder freigegeben, jedoch bleiben die Icons und alle anderen Informationen in der Desktopdatei zurück. Natürlich kann man Aladin, wie oben beschrieben, dazu zwingen die Desktopdatei neu anzulegen. Elegante Abhilfe schafft hier jedoch ein phantastisches Utility namens *DiskExpress*. DiskExpress komprimiert die Desktopdatei, packt die Harddisk und bringt die gesamte Organisation der Platte wieder auf Vordermann.

FEdit gibt´s bei der MAXON Computer GmbH in der Aladin-PD-Sammlung auf der Diskette 9, *FCopy* ist ebenfalls bei MAXON in der ST-PD-Sammlung Diskette 77 zu beziehen, DiskExpress und die SUMs sind bei jedem guten Mac Mail Order-Versand wie z.B. Pandasoft oder Gravis, erhältlich.

Als Technische Unterlagen kann ich *Inside Macintosh* von Addison & Wessley und die Aladin-Technotes wärmstens empfehlen. *Inside I-III* ist bei Pandasoft zu kaufen. Die Aladin Technotes sind im Lieferumfang von Aladin enthalten und befinden sich auf der Rückseite der Aladin-Startdiskette.

Calvi

NEU & AKTUELL



Buch incl. Programm-Diskette
Hardcover B-406

DM 59,-



Über 570 Seiten
mit Programmdisketten
Hardcover B-415

DM 59,-



ca. 300 Seiten
Hardcover B-409 DM 49,–

Programmdiskette
zum Buch D-249 DM 39,–



Über 300 Seiten
B-414

DM 49,-



Hardcover, über 430 Seiten
mit Programmdiskette
B-421

DM 69,-



Hardcover, 453 Seiten
mit Programmdiskette
B-400

DM 59,-



Hardcover
Bestell-Nr. B-436 **DM 59,-**
ISBN 3-923250-77-0
Inclusive Diskette mit Interaktiver Assembler-Entwicklungssoftware



220 Seiten – Hardcover
Bestell-Nr. B-432 **DM 49,-**
ISBN 3-923250-76-2
Inclusive Diskette mit Sicherheitssystem



Hardcover
Bestell-Nr. B-435 **DM 59,-**
ISBN 3-923250-79-7



200 Seiten – Hardcover
Bestell-Nr. B-433 **DM 29,-**
ISBN 3-923250-75-4

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ____________________

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnr. ____________________

PLZ, Ort ____________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Alle Preise sind unverbindlich empfohlene Verkaufspreise.

Heim Verlag

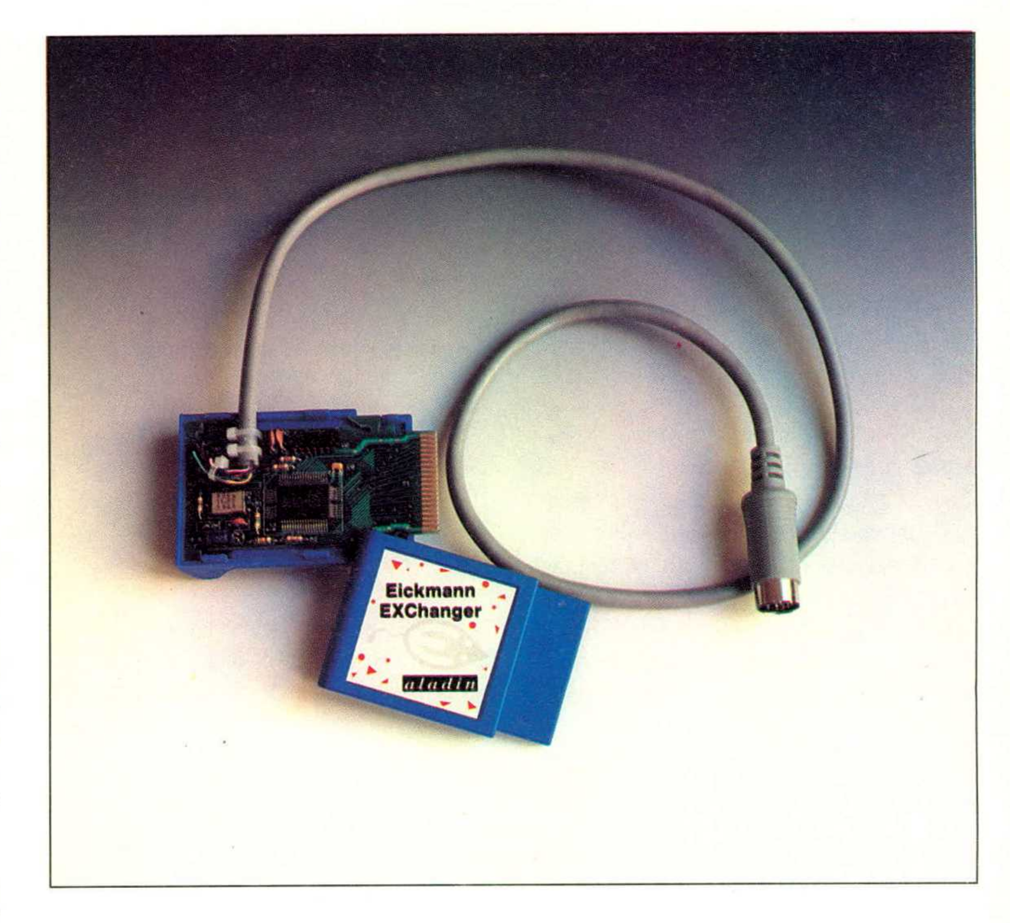
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

Spiel ohne Grenzen

Der ST liest MAC-Disketten

Ja, ab 1992 soll der EG-Binnenmarkt Wirklichkeit werden; die Grenzschranken werden fallen. Bereits heute sind allerdings schon die Schranken zwischen zwei Computertypen gefallen, die im Grunde nur zwei Dinge gemeinsam haben, nämlich die Prozessorfamilie und eine bedienungsfreundliche Oberfläche, das Desktop. Die Rede ist vom Apple Macintosh und dem ATARI ST.



Auf letzterem gibt es bereits seit längerer Zeit mehrere MAC-Emulatoren, pardon, "alternative Betriebssysteme". Das Hauptproblem als stolzer Besitzer so einer Emulation ist es geeignete Software für seinen ST-MAC zu bekommen; zu kaufen gibt es sie nämlich nur für die Original-MACs. Sogar das Diskettenformat ist zwischen ST und MAC verschieden, und hinzukommt, daß MAC-Laufwerke ihre Schreibgeschwindigkeit verändern, je nachdem, wo sie gerade auf der Diskette schreiben bzw lesen.

In den USA gibt es dafür seit längerem eine kleine Hardware, den Translator, der das für den hier in diesem unserem Lande kaum verbreiteten MagicSac im Schlaf erledigte. (Im wahrsten Sinne des Wortes, denn man kann getrost ein Nickerchen während des Kopierens einer Diskette machen.) Irgendwann im Laufe dieses Sommers soll eine neue Hardware, das sogenannte GCR, für den Spectre 128 kommen, aber warten wir's ab. Auf einer ATARI-Show in Kalifornien im April lief das Gerät einen Tag, dann war man aus technischen Gründen nicht mehr in der Lage weiter vorzuführen.

Neu auf dem Markt ist jetzt der sogenannte EXChanger der Frankfurter Firma Eickmann. Mit ihm ist es möglich, MAC-Disketten für Aladin, das alternative Betriebssystem aus deutschen Landen im holländischen Exil, hin und zurück zu konvertieren. "In ca. 3 Minuten läßt sich eine einseitige (400k)- und in ca. 6 Minuten eine doppelseitige (800k)-Diskette auf Aladin-verständliches Format bringen." So konnte man es zumindest von Eickmann Computer vernehmen.

Ausgepackt

Uns erreichte in der Redaktion ein kleines, blaues Kästchen, daß sich als Steckmodul EXChanger für den ROM-Port entpuppte, und jedermann dachte gleich, so sieht doch auch das Aladin-Steckmodul aus. Richtig, die Firma Eickmann bezieht anscheinend aus der gleichen Quelle, so daß man auch gleich optisch eine Zugehörigkeit zu Aladin feststellen kann. Zu unterscheiden ist der EXChanger von außen nur durch den Aufkleber und ein ca. 80 cm langes Floppykabel, daß in eine freie *Disk out*-Buchse gesteckt werden muß. Dazu gibt es noch eine Diskette gratis, die die erforderliche Software zum EXChangen der Daten enthält. Leider kann man den Datenaustausch nur

auf ST-Ebene also nicht unter Aladin vornehmen. Es wäre auch zuviel von Eickmann Computer verlangt, so elementar ein fremdes Programm wie Aladin umzuändern (auf computerdeutsch zu patchen), aber vielleicht nehmen sich die Aladin-Programmierer des Problems an und kreieren eine EXChanged Version.

Anzuschließen ist der EXChanger jedenfalls an jeden beliebigen ATARI ST, egal ob 520er, 1040er oder Mega, somit ist man speicherunabhängig und muß nur gegebenenfalls irgendwelche Zusatzprogramme (ACCs, Auto-Ordner-Programme etc.) entfernen. Es wird nicht zwischen Laufwerk und ST geschaltet, sondern benötigt wirklich nur eine freie Buchse. Man kann also auch mit zwei Laufwerken und dem EXChanger arbeiten, ohne andauernd ein Laufwerkstecker herauszuziehen. Das ist schon eine tolle Lösung.

Frisch ans Werk

Tja, ausgepackt, angeschlossen, schnell ein paar MAC-Disketten bespielt und los geht's. Auf der mitgelieferten Diskette ist nur ein Programm, da fällt die Auswahl nicht schwer.

Nach dem Start hat man ein simples TOS-Menü vor Augen (ein wenig GEM wie eine Menüleiste etc. wäre da nicht schlecht gewesen). Zur Wahl stehen vier Menüpunkte (Bild 1). Natürlich fühlt man sich gleich vom letzten (select /display options) angezogen, denn aus Gewohnheit oder auch Neugier interessieren immer die verborgenen Dinge. Nach Anwahl des Optionenmenüs (Bild 2) erhält man die Möglichkeit, Quell- und Zieldiskette, 400k- oder 800k-Format und den Puffermodus einzustellen. Letztere Einstellung muß man nur ändern, falls man nicht über mindestens 1 MByte Speicher verfügt. In diesem Fall kann man seine Festplatte als Zwischenspeicher nutzen.

Zurück ins Hauptmenü. Die Zwei gedrückt (transfer MAC/ALADIN), eine MAC-Diskette eingelegt und die Übertragung beginnt. Eine ganze Weile ging das auch gut, bis auf einmal eine bei Computerbesitzern gefürchtete Meldung auf dem Bildschirm erschien: READ ERROR. Gott sei Dank fiel mir ein, daß mein MEGA ST schon des öfteren Probleme mit seinem Laufwerk hatte. Also das Ganze noch einmal mit einem 1040er und siehe, es klappt problemlos. Die Erkenntnis war also: Man benötigt zum Übertragen ein absolut korrekt eingestelltes Laufwerk, sonst geht es nicht einwandfrei.

```
        Eickmann  EXChanger
MACINTOSH/ALADIN FORMAT TRANSFER
            REV 1.1
   (C) 1989 Eickmann Computer

This program transfers an entire disk
from MACINTOSH format into ALADIN
format, or from ALADIN format into
MACINTOSH format.

         SELECT ONE
   0) exit program
   1) transfer ALADIN/MAC
   2) transfer MAC/ALADIN
   3) select / display options

PRESS THE NUMBER 0-3
```

Bild 1: Das Hauptmenü

```
        Eickmann EXChanger
MACINTOSH/ALADIN  FORMAT TRANSFER

  TRANSFER CONTROL & OPTIONS MENU

...current conditions are

SOURCE DRIVE: A

DESTINATION DRIVE: A

TRANSFER SIZE: 2 sided 800K disk

BUFFER MODE: RAM BUFFER

  X) exit to transfer menu
  S) change source drive
  D) change destination drive
  T) change transfer size
  B) change buffer mode

select a letter (X S D T B)
```

Bild 2: Das Optionsmenü

Null Problemo...

... sagte ich mir und startete meinen ALADIN. Nach dem gewohnten Startablauf, legte ich eine der übertragenen Disketten ein und siehe, Aladin bot mir unverzüglich seine umfangreichen Formatiermöglichkeiten an. Nach einigem Überlegen, Herumprobieren etc. kam ich dann auf die Lösung: das Codewort hieß HFS. Fairerweise muß ich sagen, daß mir das Schicksal aller Tester zuteil wurde, die ohne Handbuch arbeiten müssen, da das Produkt noch zu neu und die Dokumentation noch in Arbeit ist. Wie mir später bei einem Rückruf bei Eickmann Computer versichert wurde, stehen all meine Probleme fein säuberlich im Handbuch aufgelistet.

Doch zurück zum HFS. Da der MAC seit geraumer Zeit schon mit einem hierarchischen Dateisystem arbeitet, hatte ich auch meine Disketten automatisch auf dem MAC unter HFS formatiert. Aladin hingegen arbeitet normalerweise mit einem nicht-hierarchischen System (MFS) und verfügt somit über keine richtigen Ordner. Er kann also HFS nicht so einfach verstehen. Abhilfe schafft hier ein kleines Programm *Hard Disk 20* (© by Apple), das in den Systemordner kopiert werden muß, damit es beim Start automatisch mitgebootet wird. Es ermöglicht HFS-Betrieb (Näheres ist in den Technotes zu finden). Einen kleinen Haken hat die Sache allerdings doch noch. Ist HFS installiert, gilt es für alle Laufwerke bis auf das, auf dem das System enthalten ist. Falls man nur eine Aladin-Partition hat, kann man das System aber auch auf der Superdisk installieren, von dort neu starten und verfügt dann über HFS. Erwähnen muß ich noch, daß der MAC 400k-Disketten im Gegensatz zu den 800k-Disketten MFS-mäßig formatiert, so daß man erstere immer lesen kann.

Mit HFS versehen ließen sich auch meine übertragenen Disketten einwandfrei lesen und weiterverarbeiten. Schon nach kurzer Ladezeit strahlten mich Programme wie PageMaker und Lightspeed C freundlich an. Ganz problemlos war auch der umgekehrte Weg, auf dem Aladin-Disketten auf den MAC übertragen wurden. Der MAC konnte die übertragenen Aladin-Disketten einwandfrei lesen.

Fazit

Wer jemals Programme für Aladin vom MAC mittels Kabel übertragen hat, der weiß, diese neue Möglichkeit hochzuschätzen. In 3 bzw. 6 Minuten lassen sich Disketten übertragen, die sonst via Kabel fast eine Stunde benötigen. Voraussetzung sind allerdings korrekt eingestellte Laufwerke (evtl. beim ATARI-Fachhändler nachjustieren lassen). Allerdings ist diese Übertragungslösung auch nicht ganz billig. Mit DM 498,- liegt der Preis des EXChangers schon über dem Preis von Aladin (ohne ROMs) selbst und ist so sicherlich nur etwas Leute mit größerem Geldbeutel oder hohem Datendurchsatz bei der Übertragung.

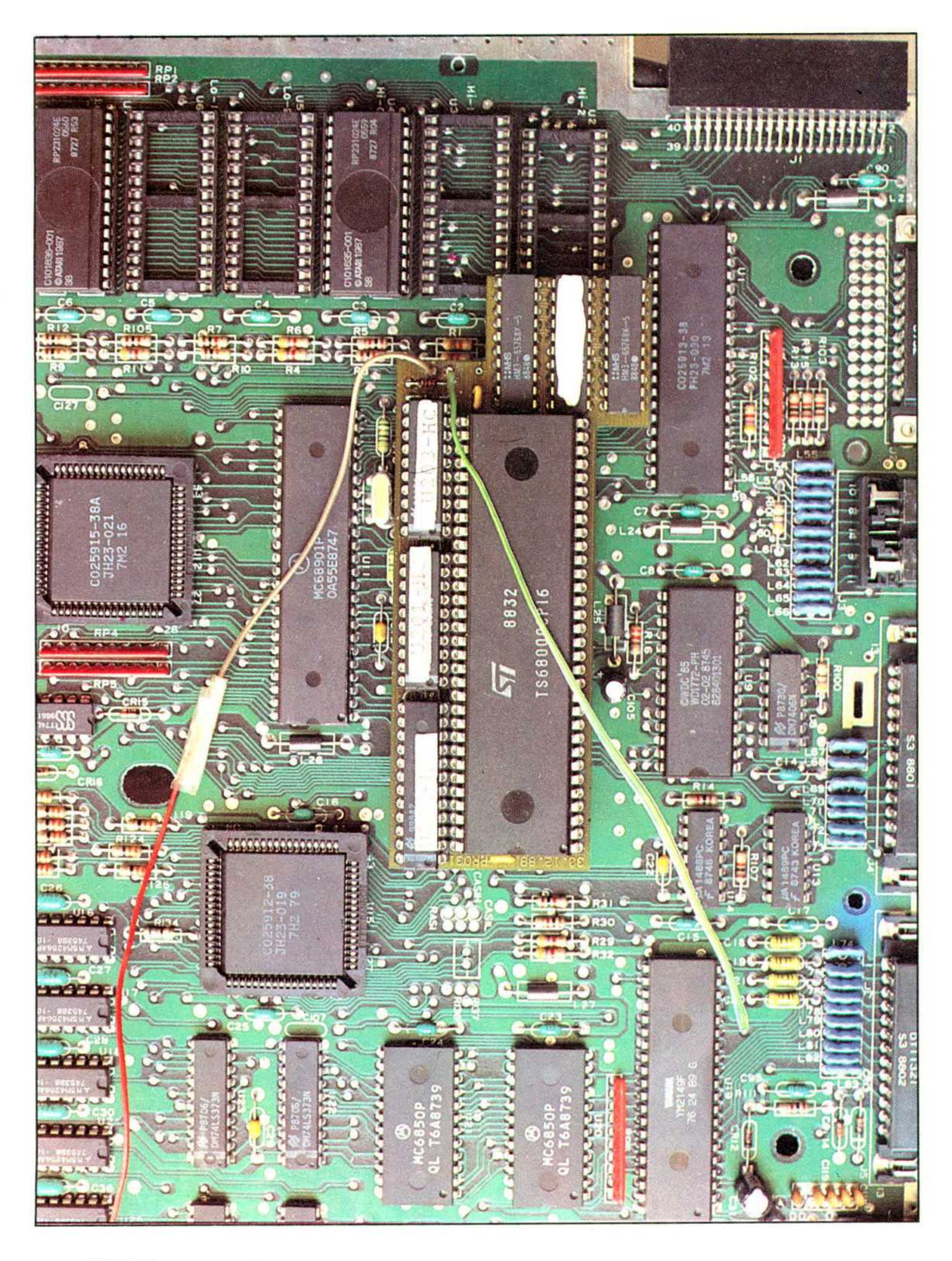
Martina Pfahl

Bezugsadresse:

Eickmann Computer
In der Römerstadt 249
6000 Frankfurt
Tel.: 069/763409

DOPPELTES TEMPO?

16 MHz Takt im ATARI ST



In einer Zeit des ungeduldigen Wartens auf eine neue Generation von ATARI-Rechnern sind alle Verbesserungen für den ST willkommen. Sinnvoll wäre, wenn solche Verbesserungen oder Erweiterungen von ATARI selbst kämen, aber da das nicht passiert ist, basteln ständig kleinere, aber leistungfähige Firmen am ST.

Die erste große und mit Sicherheit sinnvolle Erweiterung des ATARI ST war die Aufrüstung dieses Rechners mit einem M68020-Prozessor. Sie hat eine beachtliche Geschwindigkeitssteigerung bei der Ausführung vieler Programme gebracht, war aber mit verschiedenen Problemen verbunden, so daß sogar bis heute nicht alle Programme laufen. Die Erweiterung hat sich nicht nur auf eine Hardwareaufrüstung beschränkt, sondern es war auch eine Änderung des Betriebssystems notwendig. Trotz all dieser Komplikationen hätte die Aufrüstung mit Sicherheit größere Verbreitung gefunden, wenn ATARI dahintergestanden hätte. Dies war aber, wie eingangs erwähnt, nicht der Fall.

Wir beschäftigen uns in diesem Artikel mit einer ähnlichen Lösung, die aber um einiges weniger kompliziert gestaltet wurde. Einen so leistungsstarken Rechner wie den ST kann man in verschiedenen Richtungen sinnvoll erweitern. Man kann zum Beispiel seine interne Speicherkapazität erweitern, was auch verschiedene Hardwarehersteller anbieten (ST Computer-Leser der ersten Stunde werden sich erinnern, daß in unserem allerersten Heft

eine Erweiterung des damaligen ST 520 abgebildet wurde). Auch die Diskettenkapazität, die bei dem ST ziemlich knapp ist, kann erweitert werden (dies hat Claus Brod vor 2 Jahren durch sein Hyperformat erreicht). Ebenso läßt sich der grafische Bereich des ST erheblich erweitern.

Was auf jeden Fall und ohne großen Aufwand zu verbessern wäre, ist die Geschwindigkeit des M68000-Prozessors. Bekanntlich wird dieser Prozessor in drei verschiedenen Ausführungen geliefert: mit 8 MHz, 10 MHz und 16 MHz. Der Prozessor des normalen STs wird mit nur 8 MHz getaktet. Die Firma proVME hat für alle ST-Rechner eine Platine entwickelt, auf der ein mit 16 MHz getakteter M68000-Prozessor sitzt. Ebenfalls enthalten sind ein 8 kByte-Instruktions-Cache und ein 8 kByte-Daten-Cache enthält, die den Ablauf der Programme stark beschleunigen.

Der Aufbau

Wie bei fast allen fremden Erweiterungen, die man am ST vollzieht, geht es auch hier nicht ohne Löten. Dies ist die direkte Konsequenz des niedrigen Preises für diesen Rechner. Eine geschlossene Architektur ohne Möglichkeit für großartige Erweiterungen macht solchen Herstellern (und auch den Anwendern) das Leben sehr schwer. Dazu kommt noch, daß fast alle Bausteine direkt auf das Hauptplatine (Motherboard) gelötet sind.

In unserem konkreten Fall muß der Prozessor ausgelötet oder mit einem Seitenschneider müssen die einzelnen Beine des Prozessors (wie im mitgelieferten Handbuch beschrieben) abgeschnitten werden. Später kann man die Reste mit Pinzette und Lötkolben entfernen. Dann wird ein mitgelieferter Zwischensockel eingelötet, und oben drauf wird die Karte mit dem neuen Prozessor und dem Cache eingesteckt. Jetzt müssen nur zwei Drähte gelötet werden: Der eine verbindet die Karte mit dem Pin 15 des Soundchips (Ein- und Ausschalten des Cache-Speichers), und das andere führt zu Pin 39 des Shifters (dadurch wird der 16 MHz-Takt gewonnen).Wenn alles richtig gemacht wurde, können Sie den Rechner wieder zusammenbauen, die mitgelieferte Diskette in das Laufwerk stecken und neu starten.

Auf der mitgelieferten Diskette befindet



Bild 1: Das Accessory zum Ein- und Auschalten der Hypercache

Programme, die auf Hypercache ST laufen

Wie schon im Text erwähnt, laufen alle Programme, die wir getestet haben, einwandfrei. Wir konnten jedoch nicht alle für den ST existierenden Programme testen. Insofern haben wir hier eine Liste aufgeführt, die in keiner Weise vollständig ist, und hoffen, daß sich die Anwender selbst bei den Herstellern melden, falls irgendein Programm nicht laufen sollte.

Textverarbeitung, DTP:
Word Plus
Signum2!
Redakteur
Calamus

Programmiersprachen:
GFA-BASIC
OMIKRON.BASIC
Laser C
Turbo C
Megamax Modula-2
SPC Modula-2
ST Pascal plus

Datenbanken:
dBMAN V
Regent Base
Adimens

Tabellenkalkulationen:
VIP
BECKERcalc

Grafik:
Public Painter
Mega Paint II
STAD
Imagic
Creator

Emulatoren:
Aladin
PC ditto

DIAGNOSEUTILITY fuer das HYPERCACHE der

proVME
Industrielle Rechnertechnik GmbH

Pass 1
Pass 2
Pass 3

Postfach 1372
6903 Neckargemünd 1
Tel.(06223) 72029
FAX (06223) 71873

Bild 2: Ein Diagnoseprogramm überprüft den richtigen Einbau.

Benchmarks

Wir haben Hypercache mit Word Plus auf seine Geschwindigkeit getestet. Uns ist klar, daß mit so einem Programm nur schwer ein objektiver Test zu erzielen ist, aber weil gerade Word Plus eins der meist verwendeten Programme auf dem ATARI ST ist, möchten wir einen Eindruck der Geschwindigkeitssteigerung gerade bei diesem Programm weitergeben.

Wir haben einen Text mit 620 Zeilen je 80 Zeichen erzeugt und mit der Cursor-Taste gescrollt. Die Option WP-Modus sowie das Lineal am oberen Rand wurden abgeschaltet. Den Test haben wir grundsätzlich bei abgeschaltetem Blitter durchgeführt. Hier sind die Werte:

ATARI ST 8 MHz:	2:03 Minuten
ATARI ST 16 MHz ohne Cache-Speicher:	1:35 Minuten
ATARI ST 16 MHz mit Cache-Speicher:	1:15 Minuten
ATARI ST 16 MHz mit Cache-Speicher und Turbo ST:	1:05 Minuten
ATARI ST 16 MHz ohne Cache-Speicher, mit Turbo ST:	1:13 Minuten

Um den Geschwindigkeitsunterschied des STs mit und ohne Cache festzustellen, haben wir die folgenden Tests durchgeführt. Bei den ersten beiden Tests handelt es sich um Integerarithmetik, der letzte beinhaltet mehrere Multiplikationen und Divisionen mit Festkommawerten.

ATARI ST mit 8 MHz

Eratosthenes	100 Iterationen	42 s
Fibonacci	100 Iterationen	208 s
Fließkommaoperation		40 s

ATARI ST mit 16 MHz ohne Cache

Eratosthenes	100 Iterationen	42 s
Fibonacci	100 Iterationen	208 s
Fließkommaoperation		34 s

ATARI ST mit 16 MHz mit Cache

Eratosthenes	100 Iterationen	22 s
Fibonacci	100 Iterationen	104 s
Fließkommaoperation		22 s

Sieb des Eratosthenes

Manche natürliche Zahlen lassen sich als Faktoren schreiben. So gilt z.B. 8=2*4 oder 20=5*4. Es gibt aber einige Zahlen, die nicht so darstellbar sind: 17, 11, etc. Es handelt sich dabei um die sogenannten Primzahlen, wobei Eins nicht zu ihnen zählt, Zwei dagegen schon. Um solche Primzahlen unter den natürlichen Zahlen zu finden, hatte sich Eratosthenes, ein griechischer Mathematiker, der u.a. als erster den Umfang des Erdballs berechnet hat, eine Methode ausgedacht, die bis heute ihre Gültigkeit noch nicht verloren hat: Man denke sich ein Sieb, das zuerst alle Zahlen filtert, die durch Zwei teilbar sind, außer der Zwei selbst, dann die durch Drei teilbar sind, außer der Drei selbst usw. Alle Zahlen, die durch dieses Sieb durchgehen bzw. am Ende übrig bleiben, sind Primzahlen.

Fibonacci-Zahlen

Nach Leonardo Fibonacci, einem italienischen Kaufmann und Hobbymathematiker aus dem 13. Jahrhundert, sind eine bestimmte Folge von Zahlen, die heute als Fibonacci-Zahlen bekannt sind, benannt. Jede Zahl dieser Reihe (außer der Eins) wird aus der Summe der beiden vorangehenden Zahlen gebildet:

nm=nm-1+nm-2

sich ein Accessory, das zur Steuerung des Ein- und Ausschaltens des Cache-Speichers dient (Bild 1), verschiedene Benchmarkprogramme und ein Programm, das die verschiedenen Funktionen des eingebauten Caches überprüft (Bild 2). Im Prinzip braucht man auch nicht mehr.

Was läuft wie schnell?

Wie oben erwähnt, hat man schon vor längerer Zeit eine ähnliche Lösung gefunden, die von uns auch vorgestellt wurde. Bei der PAK-Karte gibt es aber einen großen Nachteil: Es laufen nicht alle Programme. Dies hängt mit dem Einsatz des M68020-Prozessors zusammen. Bei Hypercache handelt es sich um den gleichen Prozessor, nur wird dieser mit der doppelten Frequenz getaktet und hat zusätzlich einen 8 kByte großen Cache-Speicher. Sonst ist alles beim alten geblieben, insofern sollten auch alle Programme laufen. Wir haben verschiedene Programme ausprobiert (siehe Liste), und alle sind anstandslos gelaufen. Einige sind reine GEM-Anwendungen, andere laufen schlicht unter TOS. Auch Programme, die kritisch programmiert sind, haben den Test überstanden.

Schwieriger ist die Frage der Geschwindigkeitssteigerung, die der Austausch des Prozessors mit sich bringt, zu beantworten. Eine Erhöhung der Prozessortaktfrequenz bedeutet nicht, daß der gesamte Rechner schneller läuft. Der Prozessor selbst hat sich nicht geändert und braucht für jeden Befehl die gleiche Anzahl an Zyklen wie sein Bruder mit 8 MHz. Diese werden aber doppelt so schnell abgearbeitet, weshalb auch der CPU ihre Geschwindigkeit verdoppelt.

Cache

Der große Nachteil sogenannter CISK-Prozessoren ist, daß die Programmbefehle und Daten auf dem gleichen Speicher liegen und über einen gemeinsamen Bus am Prozessor landen. Der Prozessor bearbeitet einen Befehl nach dem anderen, und wenn der Bus belegt ist, muß er warten... und warten... und nochmals warten. Man hat dieses Problem mit einem kleinen, aber sehr schnellen Speicher gelöst: dem sogenannten Cache-Speicher. Er ist bei M68020 und M68030 in CPU intern eingebaut.

Bei Hypercache wird dem Prozessor ebenso ein Cache zugeordnet, hier allerdings extern. Das trägt dazu bei, daß der Ablauf (fast aller) Programme schneller

Betrachten wir die ersten Zahlen dieser Reihe:

1,1,2,3,5,8,13,21,34,.......

Die Logik der Reihe ist leicht zu erkennen.

Fließkommaoperationen

Hier das Listing dieses Testes in C. Die Ergebnisse sind sehr abhängig vom verwendeten Compiler.

```
/* Benchmarktest mit Fließkomma-
   operationen */

#include <stdio.h>
#include <osbind.h>
#include "zeit.h"

#define CONST1 3.141597e0
#define CONST2 1.7839032e4
#define COUNT  10000

main()
{

   double a,b,c;
   int i;      printf("\nFloat Test\n");

   schreibzeit();
    a = CONST1;
   b = CONST2;
   for(i=0;i<COUNT;++i)
   {
      c = a*b;
      c = c/a;
      c = a*b;
      c = c/a;
      c = a*b;
      c = c/a;
      c = a*b;
      c = c/a;
      c = a*b;
      c = c/a;
      c = a*b;
      c = c/a;
      c = a*b;
     c = c/a;
  }
  schreibzeit();
  getchar();
}
```

wird. Selbstgeschriebene Routinen erreichen eine noch erheblichere Steigerung, wenn die Programmvariablen in Registern angelegt werden.

Trotzdem ist die Frage der Geschwindigkeitssteigerung noch nicht beantwortet. Im allgemeinen kann man sagen, daß sich bei allen langsamen Programmen eine spürbare Geschwindigkeitssteigerung feststellen läßt. Bei extrem schnellen Programmen wie TEMPUS fällt sie kaum auf, im Gegensatz dazu wird z.B. 1st Word erheblich schneller. Ein gutes Komplement zu dieser Karte ist Turbo ST, der sogenannte Software-Blitter. Mit der Mischung beider Lösungen hat man einen Rechner, der um einiges schneller ist als der normale ST.

Wie wir oben erwähnt haben, würden wir uns wünschen, daß solche Erweiterungen von ATARI selbst als Standard duchgeführt werden. Das ist jedoch, egal aus welchen Gründen, nicht passiert, aber andere Anbieter bringen solche Lösungen, was wir nur begrüßen können. Wir hoffen, daß diese Erweiterung trotz des Preises (DM 590.-) zu einem Standard wird.

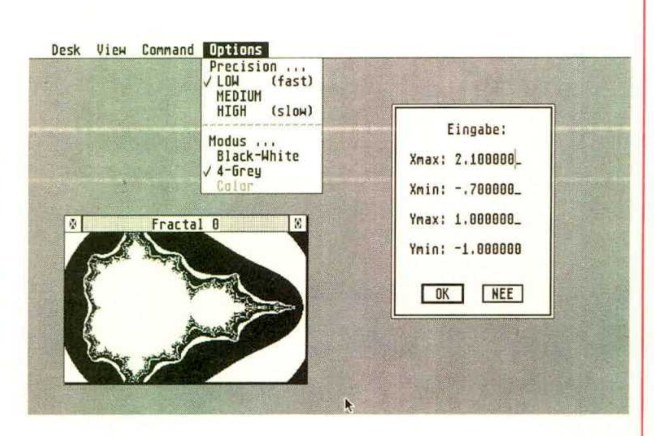
Kai Bärtels

Bezugsadresse:

Computer-Peripherie Ross
Wiener Weg 4
5000 Köln 40
Tel.: 0221/488787

Fractals

Wir haben Hypercache mit einem Fractalgenerator (ST-Public Domain-Sammlung, Diskette 100) getestet. Der hier abgebildete Fractal wurde auf einem ST mit 8 MHz und einem mit Hypercache ausgerüstetem ST erzeugt. Der Geschwindigkeitsunterschied ist eindeutig:



ST mit 8 MHz
5:00 Minuten

ST mit Hypercache
3:30 Minuten

3

STARKE SOFTWARE

Neue Version

ST LEARN
DER VOKABELTRAINER FÜR DEN ST

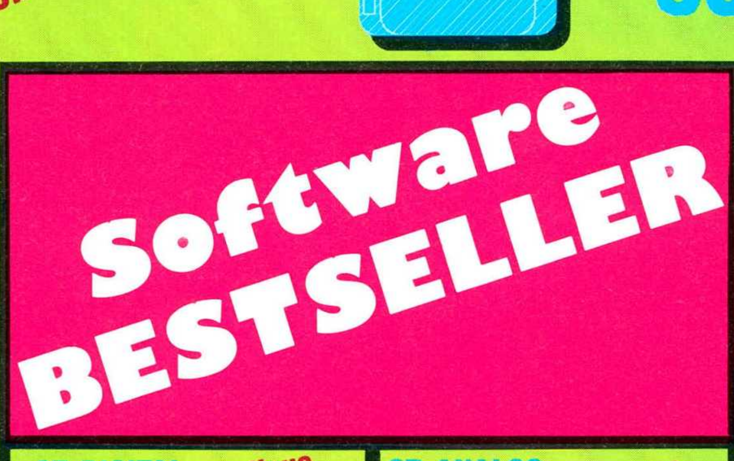
programmiert v. D Owerfeldt, Gewinner der GOLDENEN DISKETTE '87 für das beste Lernprogramm.
- Läuft vollständig unter GEM
- „Intelligente" Auswertung der Benutzereingaben
- Fehlerhäufigkeit einer Vokabel wird berücksichtigt
- Berücksichtigung mehrerer Bedeutungen eines Wortes
- Vielfältige Möglichkeiten des Lernens und der Abfrage
- Integriertes Lernspiel „HANGMAN"
- Spezielle Auswertung für unregelmäßige Verben (bei Eingabe von „to go" werden die anderen 2 Formen nachgefragt)
- Bei offensichtlicher Ähnlichkeit der Wörter wird wahlweise ein 2. Versuch zugelassen
- Trotz Einordnung der Vokabeln nach Lektionen oder Wissensgebieten ständig schneller Zugriff auf **alle** Vokabeln (nur durch Größe des Speichermediums begrenzt)
- Voller europäischer Zeichensatz (Zugriff durch die Maus unter GEM)
- Auch für Farbmonitor in mittlerer Auflösung
- Wörterbuchfunktion durchsucht alle Files eines Speichermediums nach einer Übersetzung ab
- Verbessertes Eingabeformular

ACHTUNG!!! NEU an ST-LEARN:
Allgemeines – Alle GEM-Routinen wurden überarbeitet u. optimiert, das Programm ist noch schneller geworden. – Besitzer eines SW-Monitors kommen in den Genuß einer neuen, etwas unkonventionellen (C) Mitteilung unter dem Menüpunkt **Über ST-LEARN.** – In der unteren Sonderzeichenleiste sind weitere Sonderzeichen hinzugekommen. – Im Lieferumfang befinden sich **jetzt auch mehrere englische Vokabeldateien, mit über 1.600 einfachen Grundwortschatzvokabeln**, unterteilt in 2 Schwierigkeitsstufen:
***EASY1.VOK** und **EASY2.VOK** enthalten je 400 einfachere Vokabeln.
***HARD1.VOK** und **HARD2.VOK** enthalten 800 etwas schwerere Vokabeln.
Weiterhin werden noch einige kleinere Vokabeldateien mit wichtigen Wörtern, etwa Strukturwörter oder Ordnungswörter mitgeliefert.
Eingabe- und Editierfeld – Hier ist der Knopf **'Löschen'** hinzugekommen, der es ermöglicht, die gerade angezeigte Vokabel zu löschen. Alle nachfolgenden Vokabeln werden aufgerückt.
Auswertung – Die Auswertung der Vokabeln wurde weiter verbessert, sie ist noch differenzierter geworden.
Druckeranpassung – Es ist jetzt möglich, **ST-LEARN** auf wirklich **jedem Drucker** anzupassen. Dazu gibt es im Ordner **LEARN_IT** das Konfigurationsprogramm PRINTER.
Einschaltmeldung – Hinzugekommen ist eine Einschaltmeldung, die direkt nach dem Starten des Programmes erscheint. Sie zeigt den freien Speicherplatz an.
Weitere Vorzüge von ST-LEARN – Vokabeldateien können auch nach Themen angelegt werden. Dennoch ist der Zugriff auf **alle** auf Diskette vorhandenen Vokabeln möglich. – Der **Status** einer Vokabel drückt deren Bekanntheitsgrad aus. Ist der Status z. B. kleiner als Null, so wurde dieser Ausdruck mindestens einmal nicht gekonnt etc. Je niedriger der Status, desto schlechter wurde die Vokabel nicht gekonnt und desto öfter wird sie vom Programm abgefragt.

DM 69,–

Update DM 19,–

* alle Preise sind unverbindlich empfohlene Verkaufspreise



ST DIGITAL
LOGIKSIMULATOR FÜR DEN ATARI ST

Neue Version 2.0

Ein Programm zum Erstellen, Testen und Analysieren von Logikschaltungen für Ausbildung und Hobby-Elektronik.
- Komfortable GEM-Umgebung
- Bauteile lassen sich per Maus plazieren u. verdrahten
- Umfassendes Bauteile-Set (Grundgatter, Ein-/Ausgabe-Bausteine, Flip-Flops, etc.)
- Definition zusätzlicher Bauteile durch Makrotechnik
- Makros können in Libraries gespeichert werden
- Interaktive Simulation mit Darstellung der Leitungszustände (d. h. Betätigung von 'Schaltern' mit der Maus und sofortige Reaktion der Schaltung)
- Erzeugen von Impulsdiagrammen
- Hardcopy-Funktion
- Hardwarevoraussetzung: ST/Monochrom-Monitor
- gängige Druckertreiber vorhanden

Neu ab Version 2.0

- vergrößerter Arbeitsspeicher (4 Bildschirme)
- Gruppenfunktion-Editor (Verschieben, duplizieren, löschen)
- Ausgabe von Schaltbildern und Diagrammen in Bilddatei
- erweiterte Druckfunktion (ganzes Bild im Querformat)
- zusätzliche Druckertreiber vorhanden
- **Macros:**
- Gehäusegröße in Zweierschritten wählbar
- Kleinere Anschlußsymbole
- Konvertierung der Hauptschaltung in Macros und umgekehrt
- **Simulation:**
- Berücksichtigung von Bauteilschaltzeiten
- Einzelschrittausführung mit beliebiger Schrittweite
- **Impulsdiagramme:**
- abspeichern der Eingangsdaten
- Diagrammgröße bis zu 320 Schritten wählbar
- Setzen von Marken
- **16 Bit-Interface:**
- Signal Ein-/Ausgabe
- Logikanalyse
- Bauanleitung im Handbuch

DM 89,–

Update einschl. komplett neu überarbeitetem Handbuch (nur Originaldiskette einsenden)

DM 29,–

ST ANALOG
SIMULATION VON ANALOGSCHALTUNGEN

NEU

Das Simulations-Programm zum Analysieren, Testen und Entwickeln von analogen Elektronikschaltungen (Kettenschaltungen) für Hobby, Ausbildung und Studium.
- Komfortable Maus-Steuerung
- Grafischer Schaltungsaufbau
- Einfache Eingabe und Änderung von Bauteilwerten
- Max. 65 Bauteile pro Schaltung (z. B. Stromquellen, Übertrager, Schwingkreise, offene und kurzgeschlossene Stichleitungen, Übertragungsleitungen usw.)
- „Wobbel-Generator" von 1 Hz bis 2 GHz
- Max. 0,1 Hz bzw. 1 Hz Auflösung
- Wahlmöglichkeit zwischen sehr schneller oder sehr genauer Berechnung
- Grafische Ausgabe von Spannungs- und Stromverhältnissen, von Phasenverläufen und von Eingangsimpedanzen
- logarithmische und lineare Koordinatenachsen
- Verstärkung und Offset einstellbar
- Einfaches Testen der Schaltung im Rückwärtsbetrieb
- Digitale Anzeige von Funktionswerten bei diskreten Frequenzen
- Hardcopy-Funktion
- Hardwarevoraussetzung: ST mit Monochrom-Monitor
- ausführliches deutsches Handbuch

DM 98,–

ST-MATH
DAS PROGRAMM FÜR SYMBOLISCHE ALGEBRA UND ANALYSIS

*Das Programm ST-MATH ist ein Mathematikprogramm für den ATARI ST-Computer, das es Ihnen möglich macht, symbolische Mathematik auf Ihrem Computer zu betreiben, eine für Microcomputer seltene, für den ST **einmalige** Anwendung.*
Mit den herausragenden Fähigkeiten des Programmes ST-MATH kann ein Schüler von Beginn der 8. Klasse an bis weit über das Abitur in die ersten Studiensemester vorteilhaft arbeiten. Ja, auch ein Achtkläßler kann dieses Programm bereits sinnvoll einsetzen, da die Kenntnis der höheren Mathematikfunktionen nicht Voraussetzung für das Arbeiten mit ST-MATH ist.
Also: Eine langfristige und wertvolle Anschaffung zu einem günstigen Preis.

- rechnet ohne Rundungsfehler
- verarbeitet symbolische Ausdrücke wie $2\times + 3y = 5z$
- löst Gleichungen nach beliebigen Variablen auf
- beherrscht Grenzwerte, Differential- und Integralrechnung
- ideal für Matritzenrechnung
- unglaublich schnell, da **vollständig** in Assembler geschrieben
- nicht nur ein Mathematik-Programm, sondern gleichzeitig eine komplette KI-Sprache, die leicht erlernbar ist
- die Kombination von Mathematik-System und Programmiersprache ermöglicht auch Anfängern komplexe Mathe-Programme mühelos zu schreiben
- einfache und komfortable Bedienung
- für Schüler, die sich Fehlrechnungen ersparen wollen
- für Lehrer, die mit ST-MATH Klausuren mühelos und schnell korrigieren wollen
- für Studenten, die lange Umformungen und Rechnungen zeitsparend durchführen wollen
- für Ingenieure und sonst. Anwender, die oft komplexe nichtnumerische Probleme lösen müssen
- für Jeden, der eine preiswerte, aber dennoch vollwertige Sprache für künstliche Inteligenz sucht, die leicht zu erlernen ist
- Hardware: ATARI ST mit 512 KB und ROM-TOS oder 1 MB (dann auch RAM-TOS möglich), eins. Disk.-Laufw.
- arbeitet mit Farb- und SW-Monitoren
- Lieferung mit umfangreichem deutschen Handbuch, das die Mathematikfunktion detailliert erläutert und zugleich eine ausführliche Einführung in die Programmierung von ST-MATH enthält.
- Ausführliches Informationsblatt über den Umfang und die Möglichkeiten von ST-MATH kostenfrei beim Verlag erhältlich.

Ein Spitzenprogramm DM 98,–

Achtung: Version 2.1 – Austausch

Alle ST-MATH-Besitzer können die aktuelle Version 2.1 ab sofort erhalten. Senden Sie **nur** die Originaldiskette zurück (kein Handbuch) und legen Sie für den Versand 5,– DM in Briefmarken bei.

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ______________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

Auslieferung in der Schweiz:
DataTrade AG
Langstr. 94
CH-8021 Zürich

Planen mit BECKERcalc

Seit ewigen Zeiten hilft sich der Mensch bei rechenaufwendigen Kalkulationen mit Geräten, die das intensive Rechnen stark vereinfachen. Der Einsatz des Abacus' vor 'zig Jahren oder die Tabelliermaschinen zu Anfang dieses Jahrhunderts sind Beispiele dafür, wie verschiedene Maschinen bei der Auffassung und Manipulierung von Zahlen und Daten eine große Hilfe darstellen. Der massive Einsatz von Computern bedeutete in diesem Bereich eine Revolution. Seit deren Einführung wurden eine Menge verschiedener Programme angeboten, die das Rechnen von Zahlenkolonnen automatisieren. Im Bereich Personalcomputer ist Multiplan mit Sicherheit ein etablierter Begriff. Für den ATARI ST wurden ebenso von Anfang an verschiedene Tabellenkalkulationsprogramme angeboten. Wir werden uns in diesem Artikel mit dem BECKERcalc beschäftigen, einem Spreadsheet-Programm aus dem Hause Data Becker.

Data Becker hat mit den Jahren das Aussehen seiner Software verändert. Aus dem weißen Hintergrund mit roter Schrift wurde jetzt weiße Schrift auf rotem Hintergrund. Nichts auf dem Cover des Schubers verrät, daß sich innen eine Tabellenkalkulation befindet - bis auf den Firmennamen und den großen Schriftzug 'Software'. Sieht nicht schlecht aus. Man hat gelernt, daß auch Anwender 'mit den Augen essen'. Aber was gut aussieht, muß nicht unbedingt gut schmecken, deshalb vergessen wir die Äußerlichkeiten und probieren lieber ein Stück GEM-Menü.

Ein Tabellenkalkulationsprogramm ist nicht für jedermann zu gebrauchen. Wenn sie nur wenige Werte zu addieren haben, sind Sie sicher mit einem Bleistift und einem Stück Papier besser bedient als mit irgendwelchen Programmen. Sollen aber mit einer großen Anzahl von Werten nicht nur Summen, sondern auch verschiedene und aufwendige mathematische oder wirschaftliche Operationen durchgeführt werden, ist eine Tabellenkalkulation ein besseres Werkzeug als die Kombination von Bleistift und Papier. Damit stellt sich die erste grundlegende Frage: Was soll ein solches Programm können? Die zweite Frage ist: Was kann BECKERcalc? Ich gebe zu, daß man den Fragenkatalog auch anders gestalten könnte. Zum Beispiel: Was brauche ich? Und dann die andere Frage, die nächste usw. Aber das könnte dazu führen, alles so zu relativieren, daß wir irgendwann zu dem Schluß kämen, daß dieser Bericht keinen Grund hat, geschrieben zu werden. Aber so weit möchten wir es nicht treiben, sondern fangen wir endlich an.

Das Fenster zum Hof kann rechnen

Es handelt sich hier um GEM und Programme, die diese umsonst mitgelieferte Bedienungsoberfläche verwenden. Bei BECKERcalc wurde von der leistungsstarken Oberfläche reichlich Gebrauch gemacht, was in meinen Augen auch richtig ist, da schon zu viele Softwarehersteller ihr eigenes Süppchen kochen.

Das Programm präsentiert sich in bester GEM-Manier mit vielen Icons und Pull-Down-Menüs. Die Arbeitsblätter, die Daten aufnehmen sollen, sind Einzelfenster, die man sowohl verschieben als auch in ihrer Größe verändern kann. Mehrere dieser Arbeitsblätter kann man gleichzeitig verarbeiten, wobei jedes eine maximale Kapazität von 200 Spalten und bis zu 1000 Zeilen hat. Verwenden Sie aber gleichzeitig mehrere Arbeitsblätter, kann es vorkommen, daß Ihr Arbeitsspeicher für diesen Aufwand nicht ausreicht.

Ein Arbeitsblatt ist in viele Felder aufgeteilt, die man mit Text, Zahlen oder Formeln belegen kann. Die Felder lassen sich

ebenso in verschiedenen Modi manipulieren. Zum Beispiel kann man eine ganze Spalte in der Breite vergrößern oder verkleinern oder für ein Feld alle Textausgaben zentrieren lassen. Die Formatoption beherrscht wichtige Merkmale für die Nutzung dieser Tabelle in verschiedenen Bereichen. So kann man unter anderem zwischen Integer- oder Festkommadarstellung (hier kann zwischen einer, zwei, drei oder vier Stellen nach dem Komma ausgewählt werden), Exponentialdarstellung, Währungssymbol (man kann beliebige Währungssymbole eingeben), Datum- und Zeitdarstellung wählen. Möchte man eine eingegebene Formel vor einem versehentlichen Löschen schützen, kann dies ebenfalls unter der Formatoption geschehen. Damit hätten wir das Minumun von dem, was eine Tabellenkalkulation kann. Bis hierher unterscheidet sich BECKERcalc auch nicht besonders von anderen Programmen dieser Art. Kulinarisch ausgedrückt, sind wir bei der Vorspeise. Nach einer Pause, in der alles gut verdaut wird, machen wir mit der Hauptspeise weiter.

Desk Arbeitsblatt Bereich Edit Format Fenster Optionen

E:\BECKER.CAL\DARLEHEN(1)

Aktuelles Feld: E1

NORM	A	B	C	D	E	F	G	H	I	J
1					DARLEHEN					
2	Angebot		1		2		3			
3										
4	Darlehn in DM		50000		50000		50000			
5										
6	Zinzen % p.A.		6.75		7.25		5.75			
7	Gebühr 1mal		680		480		850			
8	Laufzeit MON.		36		36		48			
9										
10	Raten p.M.		1538.146		1549.576		1515.439			
11										
12										
13										
14										

Tabelle Drucker Charts Convert Aufnahme Makro Bereiche Fenster Hilfe

Bild 1: Das Arbeitsblatt von BECKERcalc



Bild 2: Die Druckdialogbox

Von Formeln, Bereichen, Grafiken und vom Drucken

Sie werden sicherlich fragen: "Wieso werfen die alles in einen Topf?". Na ja, damit es besser schmeckt. Aber Spaß beiseite. Dies sind wichtige Teile eines solchen Programmes, und sie müssen so flexibel sein, daß jeder Anwender sie für seinen eigenen Fall so konfigurieren kann, daß ein richtiges und problemloses Arbeiten ermöglicht wird. Betrachten wir uns die verschiedenen Bereiche, in denen es zum Einsatz kommen könnte, müßten wir zu dem Schluß kommen, daß es eine riesige Menge von eingebauten Formeln bräuchte, oder daß es so gebaut sein müßte, daß durch Kombination einzelner Operationen die gewünschte Formel erreicht wird.

BECKERcalc verfügt über eine große Menge von fest eingebauten Funktionen, die folgendermaßen aufgeteilt sind:

Grundfunktionen: Hier findet man die üblichen Operationen wie Addition, Subtraktion, Multiplikation und Division. Es sind auch die Vergleichsfunktionen eingebaut sowie andere Funktionen wie Prozentrechnung, Modulo, Fakultät, Exponential, Quadratwurzel etc. Alle aufzulisten, würde den Rahmen dieses Berichtes sprengen, hier nur einige von ihnen. Klammerrechnung sowie Rechenhierarchie beherrscht BECKERcalc ebenfalls.

Trigonometrische Funktionen: Auch hierbei sind die üblichen und meist gebrauchten Funktionen enthalten. Man muß aber nebenbei darauf hinweisen, daß für die meisten Anwender diese Funktionen nicht sehr wichtig sind.

Bereichsfunktionen: Bei diesem Teil der Funktionsbeschreibung komme ich ziemlich ins Schleudern. Zum einen weiß ich nicht genau, was ich mit dem Begriff "Bereiche" eigentlich darstellen wollte, und zum anderen, weil ich die hier enthaltenen Funktionen zumindest teilweise als statistische Berechnung einordnen könnte. Sie werden mir sicher Recht geben, wenn Sie sich einige der Funktionen anschauen: Varianz, Standardabweichung, Maximum, Minimum etc.

Finanzfunktionen: Das sind sicherlich mit die wichtigsten Funktionen von BECKERcalc, und gerade in diesem Bereich wird eine Tabellenkalkulation am häufigsten gebraucht. Eine Fülle von Funktionen zur Berechnung von beispielsweise Darlehen, Renten, Zinsen und Cashflows stehen dem Anwender zur Verfügung.

Logische Funktionen: Bis auf wenige Befehle, die sich aus der Überprüfung ergeben, ob ein Feld numerisch oder text-

Desk Grafik Vorlage Skalierung Optionen Manuell

NAMENLOS (2): TEST — 20000, 10000, 0 — Januar, Oktober — TOTAL — Y-Achse, X-Achse

NAMENLOS (3): TEST — 20000, 10000, 0 — Januar, September — TOTAL — Y-Achse, X-Achse

E:\BECKER.CAL\TEST4 (1)

Aktuelles Feld: C6

NORM	A	B	C	D	E	F	G	H
2	Januar		9478.00					
3	Februar		9087.10					
4	März		10567.00					
5	April		11089.00					
6	Mai		9673.00					
7	Juni		7109.00					
8	Juli		6975.34					

Bild 3: Es lassen sich Diagramme und das Arbeitsblatt gleichzeitig darstellen.

mäßig belegt ist, sind auch hier die üblichen Funktionen wie Oder, Und, Falsch etc. vorhanden.

Außerdem gibt es Zeit- und Textfunktionen, die die gesamte Arbeit erleichtern. Ich glaube, nicht zuviel verprochen zu haben, als ich oben erwähnte, daß BECKERcalc eine ausreichende Menge an Funktionen zur Verfügung stellt.

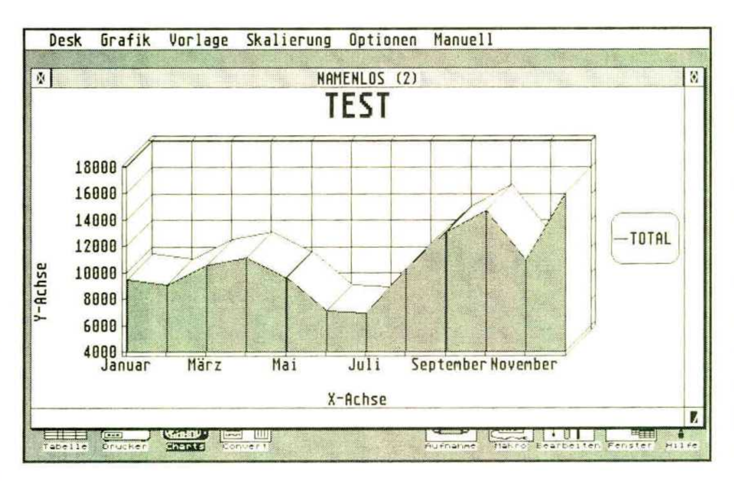
Wir haben bei der Darstellung der verschiedener Funktionen über Bereiche gesprochen. Da diese bei BECKERcalc eine große Rolle spielen, werden wir uns sie aus der Nähe betrachten. Bei einem bestehenden Arbeitsblatt kann man eine gewisse Menge an Spalten und Zeilen markieren und als Bereich definieren. Diesen kann man kopieren, verschieben, auf ein anderes Arbeitsblatt übertragen oder damit eine Grafik erstellen.

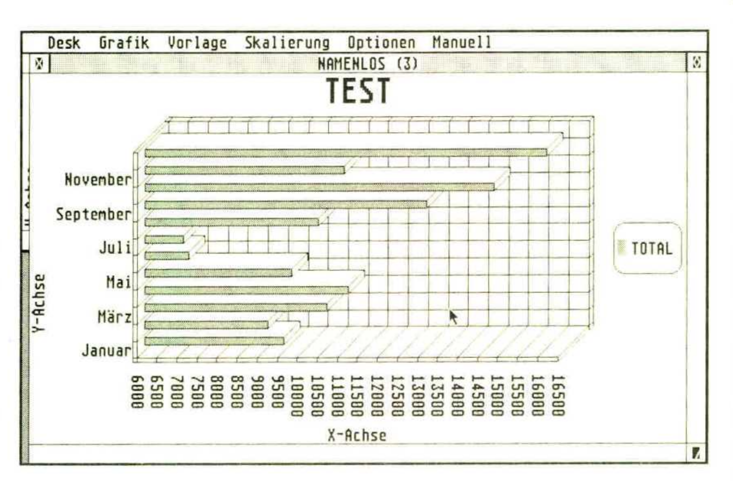
Der zuletzt genannte Bereich ist bei BECKERcalc sehr wichtig. Dabei wäre es aber nicht sinnvoll, von einem ganzen Arbeitsblatt eine Grafik herzustellen, weshalb man einen oder mehrere bestimmte Bereiche ausschneidet.

Auch für die grafische Darstellung gibt es bei BECKERcalc eine Fülle von Funktionen. Man kann prinzipiell zwischen 2D- und 3D-Darstellungen wählen, neun verschiedene Bereiche lassen sich gleichzeitig darstellen (ob das sinnvoll ist, bleibt jedem selbst überlassen) und fünf verschiedene Arten von Grafiken kann man insgesamt darstellen:

- Balkengrafik
- Liniendiagramm
- Tortengrafik
- Funktionsgraph
- Hilo-Chart

Ferner läßt sich die erstellte Grafik manipulieren. Die verwendete Skalierung läßt sich ändern, Texte, Rahmen oder Fläche lassen sich einfügen, einzelne Objekt können gelöscht werden etc. Ebenso läßt sich die fertige Grafik ausdrucken. Der Ausdruck erfolgt über das Menü "Drukken-Ausgabe" auf Metafile. Für eine Grafik, die im Metafile-Format vorliegt,





braucht man das Programm GDOS. Es ist im Lieferumfang enthalten und sollte jedesmal mitgebootet werden, wenn man mit BECKERcalc arbeiten und eine Grafik ausdrucken möchte.

Will man eine Tabelle ausdrucken, kann, aber muß dies nicht über GDOS erfolgen. Wenn Sie das Drucker-Icon wählen, erscheint ein neues Poll-Down-Menü, und Sie können jetzt einen markierten Bereich entweder als Metafile ausdrucken oder auf einem gewöhnlichen 24-Nadeldrukker. Es werden verschiedene Druckeranpassungen mitgeliefert. Es existiert aber auch die Möglichkeit, sich eine Anpassung für seinen eigenen Drukker selbst herzustellen. Ein extra Druckerprogramm mit dem Namen Pretty-Printer erlaubt unter anderem den Ausdruck im Querformat. Wie Sie wahrscheinlich bemerkt haben, sind die Druckmöglichkeiten bei BECKERcalc sehr vielfältig.

Automatisierung durch Makros

"Makros?" werden Sie vielleicht fragen, "wozu brauche ich Makros, wenn ich nicht programmieren möchte?" Ja, das mag in vielen Fällen vielleicht auch richtig sein. Aber was machen Sie, wenn Sie ein sehr aufwendiges Datenblatt mit vielen Zahlen und sehr vielen Parametern haben, die sich ständig ändern? Dann müssen Sie immer suchen, wo Ihre neuen Daten eingegeben werden müssen, und wenn Sie mal vergessen haben, wohin sie gehören...

Man kann sich aber alles ein wenig erleichtern, wenn man eine Tabelle durch ein Makro automatisiert. Ein Makro ist eine Folge von Befehlen, die man beliebig oft wiederholen kann. Es läßt sich sehr einfach erstellen, und man braucht keine besonderen Programmierkenntnisse. An den Anfang eines jeden Makros gehört ein sogenannter 'Makrokopf', der es als solches definiert. Dahinter kommen die Befehlsfolgen, die man für die Berechnung des Arbeitsblattes benötigt. Auch hier stehen dem Anwender wieder eine Fülle von Befehlen zur Verfügung, mit denen man fast jede denkbare Aufgabe lösen kann.

Zum Starten eines Makros braucht man nichts anderes zu tun, als die Icons, die mit dem Namen Makro versehen sind, anzuklicken. Es erscheint eine Tabelle mit allen vorhandenen Makros, und durch ein weiteres Anklicken des gewählten Makros startet der gewünschte Ablauf.

Fazit

Wie oben erwähnt, ist die Berechnung großer Zahlenkolonnen neben der Text-

verarbeitung mit Sicherheit eine der am häufigsten vorkommenden Aufgaben, die ein Rechner zu erledigen hat. Logischerweise sucht der Anwender das Programm, das am meisten anbietet. BECKERcalc bietet mit seiner Menge an Funktionen, Möglichkeiten und Bedienungsfreundlichkeit ein sehr günstiges Arbeitsfeld.

Eine Sache hat mich allerdings gestört: Bei jeder Eingabe muß die Return-Taste gedrückt werden, statt dafür die Pfeiltasten zu benutzen. Bei BECKERcalc muß man nämlich Return drücken und zusätzlich eine der Pfeiltasten, um in ein anderes Feld zu gelangen. Wenn man sich daran gewöhnt hat, ist es nicht mehr so tragisch (wie alles, an das man sich gewöhnt), aber am Anfang bedeutet es schon eine Umstellung.

Das Handbuch ist jedoch sehr positiv aufgefallen. Es beinhaltet ausführliche Informationen für Anfänger wie für Fortgeschrittene. Viele Beispiele in verschiedenen Bereichen helfen dem Anwender, sich einen besseren Überblick zu verschaffen.

Das Programm ist durch die Möglichkeiten, Dateien zu importieren in der Lage, Daten von Lotus 1-2-3 und Multiplan zu verarbeiten. So hat man mit diesem Programm auch die Kompatibilität zu anderer Software, die schon lange auf dem Markt etabliert ist. Wenn Sie auch vor der Aufgabe stehen, oben beschriebene Kalkulationen durchzuführen, empfehle ich Ihnen, sich dieses Programm anzuschauen.

Jutta Krill

ATARI MACHT(S) L(E)ICHT

Lichtgewitter. Fast könnte man meinen, man wäre im vierten Teil der Star-Wars-Trilogie gelandet. Bunte Laserstrahlen huschen an einem vorbei. Nur fehlt das Krachen und Dröhnen von Explosionen, die wir aus dem Weltraumepos kennen. Dafür klingt aus Lautsprechern Musik.

Die Erklärung ist einfach: Wir befinden uns nicht irgendwo zwischen Beteigeuze und Aldebaran sondern in einer Discothek im irdisch-bayrischen Voralpenland. Zusammengeführt hat uns, wie könnte es anders sein, ein ATARI ST, oder besser 3, die hier zur Grundsteuerung der Lichtanlage und seit kurzem auch als Steuersystem für die hauseigene Laseranlage eingesetzt werden.

Etwas versteckt liegt die Discothek schon, aber wenn man sie betritt, fallen dem interessierten Betrachter sofort einige Besonderheiten auf. Da ist zunächst die Ausgestaltung des Innenraumes mit Königsloge und Wandbemalung. Eine Grotte und ein Restaurant hinter Glas, von dem man einen guten Blick auf die Tanzfläche hat, stechen auch hervor.

Für den Computerenthusiasten ist aber die Musiktheke von besonderem Interesse, hinter der Disk- und Lightjockey verschwinden. Man kann den beiden bei der Arbeit über die Schulter schauen, was auch nicht alltäglich ist. Auf dem Tisch des Lightjockeys repräsentiert ein großes Display hinter Rauchglas mit LEDs die einzelnen Lichtkörper im Diskothekenraum.

Und dann sind da noch drei Computermonitore. Auf einem ATARI läuft die Grundsteuerung für die Lichtanlage, die den Lightjockey ein wenig entlasten soll. Für den Laser stehen zwei vollkommen unabhängige Computersysteme zur Verfügung. Eines wird zum Abspielen fertig eingekaufter 'Konserven'-shows benutzt, die auch schon von hoher Qualität sind, aber das eigentlich neue an dem installierten System ist die Steuerung über den zweiten ATARI. Mit ihr wird es möglich, auf einem Grafiktablett eigene Objekte zu definieren und dann zu animieren. Dazu ist natürlich ein gewisser Aufwand nötig.



Vom Strahl zum Bild

Das Laserrohr, ein Krypton-Weißlicht-Laser, hat eine Energieaufnahme von ungefähr 50 kWh bei einer Lichtleistung von etwa 2 Watt. 50 kW entspricht der Energieaufnahme von 500 100 Watt-Glühbirnen. Das Rohr wird wassergekühlt. Da Laserlicht gefährlich für das Auge sein kann, muß eine solche Anlage durch den TÜV abgenommen werden. Auch muß Vorsorge getroffen werden, daß nie die volle Leistung in einem Strahl enthalten sein kann. Dazu wird der Lichtstrahl des Lasers durch Spiegelsysteme gestreut und abgelenkt. Am Strahlautritt

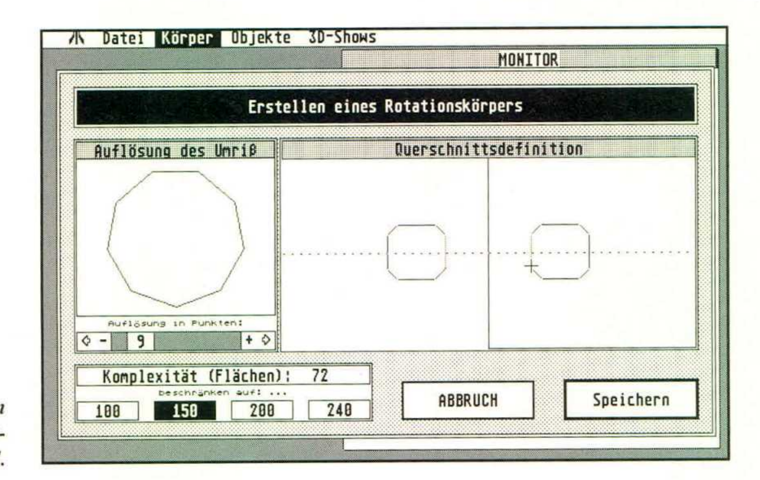


Bild 7: Definition eines Rotationskörpers im 3D-Teil.

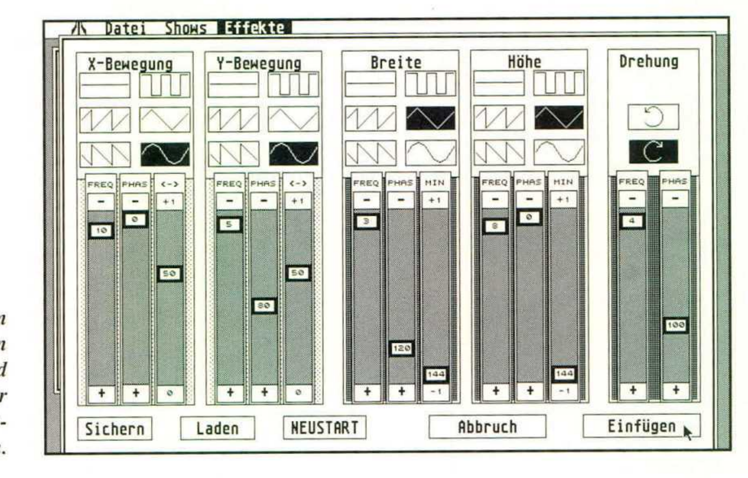


Bild 8: Mit diesem 'Mischpult' lassen sich die Bahn- und andere Effekte für Objekte aus dem 2D-Editor festlegen.

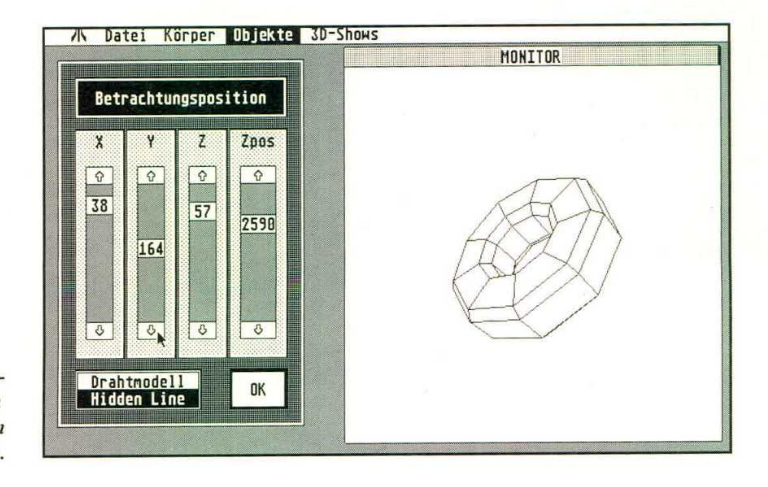


Bild 10: Berechnungen an einem 3D-Körper. Hier im Hidden-Line-Modus.

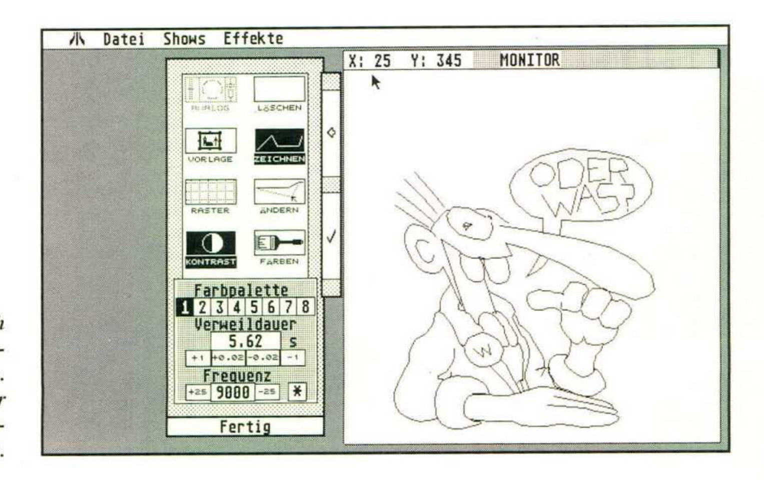


Bild 12: Natürlich sind auch komplexere Bilder möglich. Hier ein bekannter Vertreter des öffentlichen Lebens.

sitzt eine Einheit, die für die Strahlen- und Showeffekte sorgt. Sie besteht aus mindestens 3 sogenannten Scannern. Das sind halbdurchlässige Spiegel aus Spezialglas, die auf galvanometerähnliche Schrittmotoren montiert sind. Ein Spiegel hat eine Fläche von ca. 2-3 cm2, wobei allein das Glas ungefähr 400-500 DM kostet. Ein Scanner lenkt den Strahl in horizontaler (X-) Richtung ab, der zweite in vertikaler (Y-) Richtung und der dritte Scanner erledigt das 'Blanking', das Ausblenden des Strahles zwischen zwei Objekten. Problematisch ist an der Sache aber, da es sich bei den Scannern um mechanische Teile handelt, die eine gewisse Eigenträgheit besitzen. Aus diesem Grund können nur relativ enge Winkel (ca. ±30 Grad) eingeschlossen werden.

Damit der weiße Strahl Farbe bekommt, muß noch ein Farbgebungselement (Color-Box) in den Strahlenverlauf eingebaut werden. Hier gibt es zwei Methoden. Eine (relativ) billige langsame, bei der Farbfilter in den Lichtstrahl gezogen werden, und eine teure schnelle. Bei der zweiten Methode wird der Strahl in drei Strahlen aufgeteilt, die ähnlich wie beim Farbmonitor oder Fernseher den Rot-, Grün- und Blauanteil darstellen. In jeden dieser drei Strahlen, wird wieder ein Scanner eingesetzt, der den Strahl aber nur teilweise ausblendet. Nachdem so die RGB-Anteile definiert sind, wird der Strahl wieder vereinigt und in die Ablenkeinheit geschickt. Das zweite System erlaubt, je nach Regelbarkeit der Scanner, viele Farbabstufungen. Bei 256 Stufen je Scanner, solche Werte sind heute schon üblich, sind das 2563 Farbstufen, also über 16 Millionen. Hingegen erlaubt das erste System nur soviele Farben wie Filter vorhanden sind, also ungefähr 8 Farben. Mehrere Filter in den Strahl zu ziehen wäre zwar technisch möglich, ist aber nicht sinnvoll, weil dann der Strahl zu schwach und damit unsichtbar würde. Und wen interessiert schon die Farbe von etwas, das er sowieso nicht sehen kann.

Ein bißchen Rechnen

Damit aber nun ein Bild entsteht, muß erst einmal ganz schön gerechnet werden. Die Anlage läuft momentan mit einem MEGA ST 4, einem 1040er und einer Festplatte. Dabei wird der 1040er nur als Signalprozessor für die Scanner und die Color-Box eingesetzt. Die Hauptrechenarbeit findet im MEGA ST statt. Um mit dem Laser ein Bild zu erzeugen, braucht man Vektorgrafiken. Die Scanner sind

mechanische Teile und leider sehr viel träger als die elektromagnetischen Gegenstücke im Fernseher oder Monitor, weswegen pixelorientierte Grafiken unmöglich sind. Außerdem muß jedes Bild mindestens 10-20mal pro Sekunde neu gezeichnet werden, da man sonst gar nichts oder nur ein sehr stark flackerndes Bild sieht. Dank Assemblerprogrammierung in den zeitkritischen Routinen kann der ATARI mehr als 9000 Punkte pro Sekunde ansteuern, also insgesamt 9000 Linien zeichnen. Bedenkt man das jeweilige Neuzeichnen, kann man von etwa 500-700 Linienzügen ausgehen, die ein Bild enthalten darf. Damit sind schon sehr komplexe Bilder möglich. Anpassungen an runde Linienzüge nimmt das Programm übrigens automatisch vor, wodurch man einen Kreis auch mit relativ wenig Punkten erzeugen kann. Die beiden Rechner teilen sich die Arbeit sehr gut. Der MEGA ST liefert optimierte Linienzugfolgen, d.h. jedes Bild wird mit der minimalen Anzahl an Linien dargestellt, und der 1040er setzt dies in Pegel für die Scanner um und gibt sie über einen qualitativ extrem hochwertigen DA-Wandler an die Scanner weiter. Der DA-Wandler ist eine Eigenentwicklung der Hersteller und zeichnet sich durch außerordentlich kurze Reaktionszeiten und hohe Genauigkeit aus.

Grafikshow vs...

Nachdem wir nun wissen, wie ein Bild aus dem Rechner kommt, stellt sich die Frage, wie wird es hineingebracht. Dazu gibt es mehrere Möglichkeiten. Das Programm verfügt über einen 2D- und einen 3D-Editor. Im 2D-Editor kann man z.B. mit einem Grafiktablett Vorlagen abzeichnen und so in den Computer eingeben. Die so eingegebenen Vorlagen werden dann erst einmal linienoptimiert und können dann weiterbearbeitet werden. So kann jeder Linie eine eigene Farbe gegeben werden. Diese Farbgebung erfordert eine extreme Koordination von Zeichnen und Farbgebung und ist hier zum ersten Mal in dieser Art realisiert worden.

Im 3D-Editor kann man ähnlich wie bei CAD 3D ein 2-dimensionales Objekt erzeugen und diesem dann eine Tiefe geben. Dabei kann ein solches 3D-Objekt entweder als Drahtmodell oder mit Hidden-Lines (unsichtbare Linien werden nicht gezeichnet) dargestellt werden. Der Hidden-Line-Algorithmus ist einer der besten, die auf dem ST bisher realisiert worden sind, allerdings leidet er unter der

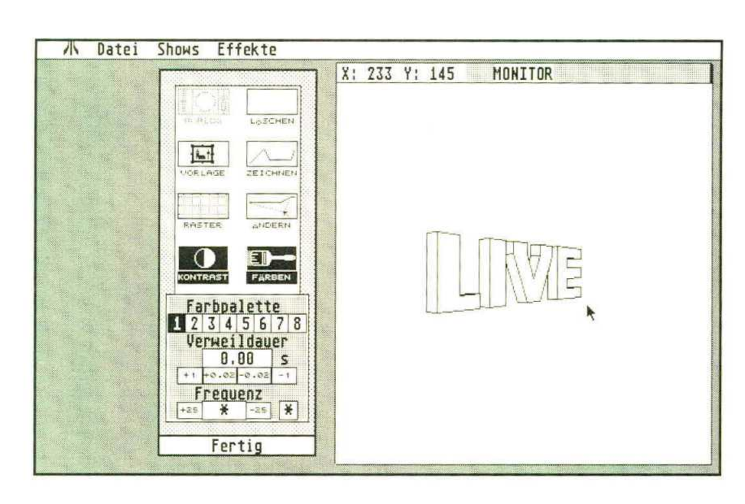
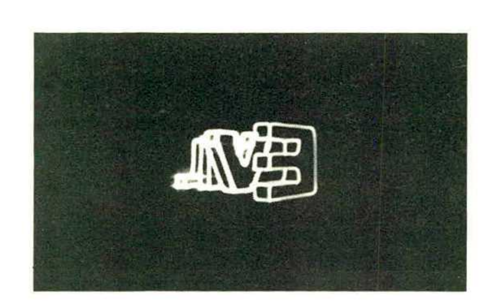
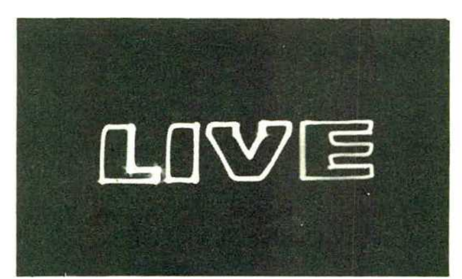
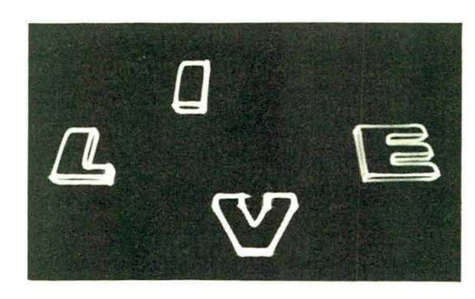
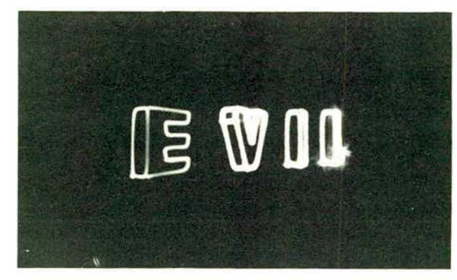


Bild 6: Arbeiten im 2D-Editor (dazu Fotoserie "LIVE")









zu geringen Rechengenauigkeit des ST. Die gesamte Arithmetik ist übrigens reine Integerarithmetik.

Um nun eine Grafikshow zu erzeugen, kann man entweder einzelne Objekte nehmen und eine Bahn für sie definieren, wobei auch die Größe des Objekts verändert werden kann, oder aber man arbeitet wie bei einem Film, indem man mehrere Bilder aneinanderhängt.

Strahlenshow

Nun kommt aber eine Tatsache ins Spiel, die man zunächst gar nicht so recht glauben mag, sobald man aber eine Vorführung erlebt hat, kann man es verstehen. Matthias, der die Hardware für das Steuerungssystem realisiert hat, formuliert das so: "Der Programmieraufwand steht reziprok im Verhältnis zur Spektakularität der Effekte." Damit spielt er auf den Aufwand für die Grafik- und die Strahlenshows an.

Das System ist nämlich in der Lage, ohne teure und empfindliche Spiegelbank Strahleneffekte zu erzeugen. Bei herkömmlichen Systemen kann so eine Spiegelbank locker einige zehntausend Mark kosten. Da es sich dabei um empfindliche, mechanisch bewegte Glasteile handelt, sind auch die Kosten während des normalen Betriebs nicht zu verachten. Bei der ATARI-Steuerung fallen diese zusätzlichen Kostenpunkte einfach weg, und der Aufwand für Pflege und Instandhaltung des Systems wird auf ein Minimum reduziert.

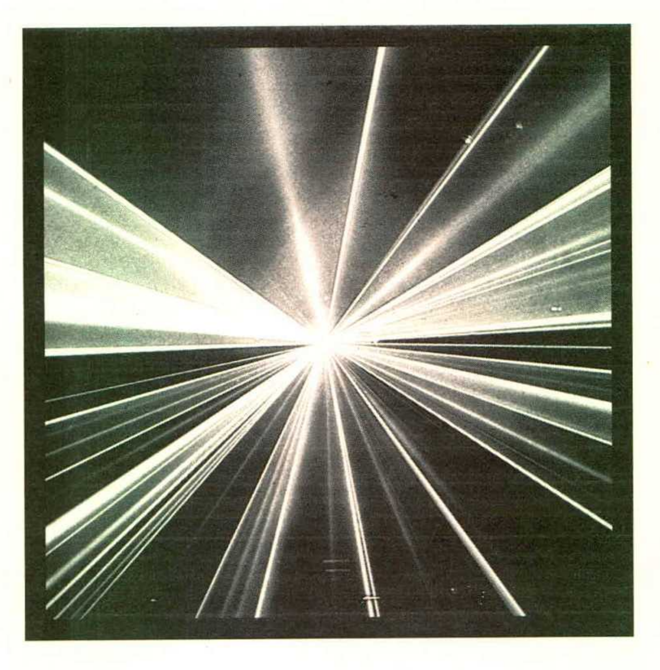
Durch geschickte Anordung von Spiegeln und anderen reflektierenden Elementen wird bei einer Strahlenshow normalerweise der gesamte Diskothekenraum ausgenutzt, während bei den vorhin erklärten Grafikshows meistens auf einer Leinwand gearbeitet wird. Damit die Strahlenshows allerdings wirken, muß unbedingt Nebel vorhanden sein. Das Licht bricht sich an den kleinen Tröpfchen und wird so erst richtig sichtbar. Das Ergebnis sind dann Tunneleffekte, breitgespannte Fächer und anderes.

Laufschriften

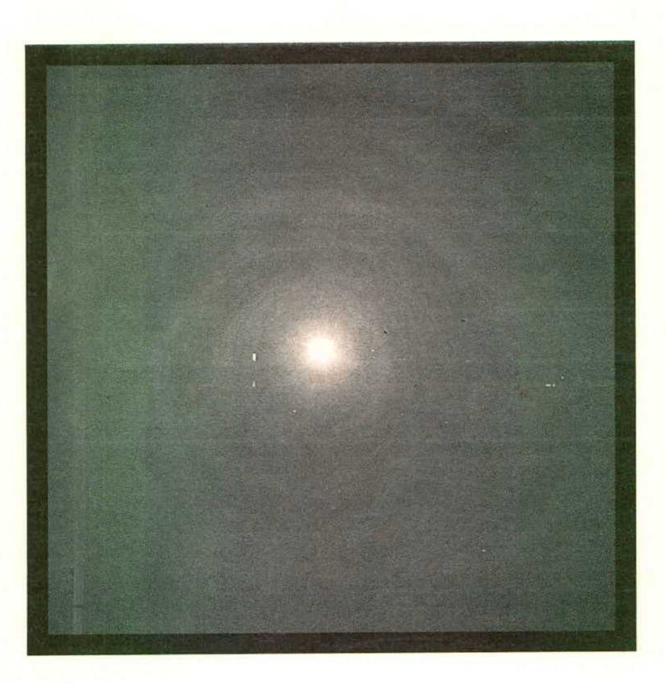
Mit dem Laser lassen sich auch Laufschriften erzeugen, wobei man sich Zeichensätze selbst erstellen kann. Verschie-

dene Textattribute sind auch möglich.

Im üblichen Betrieb wird die Anlage meist über ein anschließbares Keyboard betrieben, daß mit frei definierbaren Effekten belegbar ist. Dadurch wird letztendlich der Mensch wieder zum entscheidenden Faktor, denn ohne einen guten Lightjockey nutzt die tollste Lichtanlage nichts. Danke für die Vorführung, Andy.

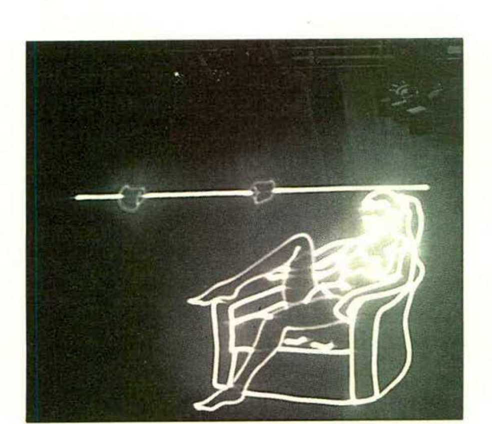


Natürlich darf die Entwicklung eines so komplexen Systems wie der Lasersteuerung nicht stillstehen, denn die Konkurrenz schläft nicht. Die Pläne für die Zukunft sind zahlreich und erste Schritte sind bereits wieder unternommen worden. So soll der 1040er gegen einen echten Signalprozessor ausgetauscht werden, ähnlich dem, der im NeXT-Rechner von Steve Jobs für den Datenaustausch zuständig ist. Desweiteren wird an einem Programm gebastelt, mit dem HPGL-Dateien eingelesen werden können. Damit wäre eine Brücke zu CAD-Systemen geschlagen, mit deren Hilfe sich dann Shows in Industriequalität schnell entwickeln lassen. In Planung ist auch eine Möglichkeit zur Umwandlung von Pixelgrafiken in Vektorgrafiken mit Übernahmemöglichkeiten von einem Videodigitizer.



Wir dürfen gespannt sein, wie sich dieses System durchsetzt; wer weiß, vielleicht sehen wir demnächst öfter einen ATARI, wenn wir in die Disco gehen. Auf jeden Fall möchte ich mich bei Matthias, Jens, Andy und dem Team vom 'Luggi' herzlich für die Vorführung bedanken, und bei Matthias für die technischen Hinweise, die diesen Artikel überhaupt erst möglich gemacht haben.

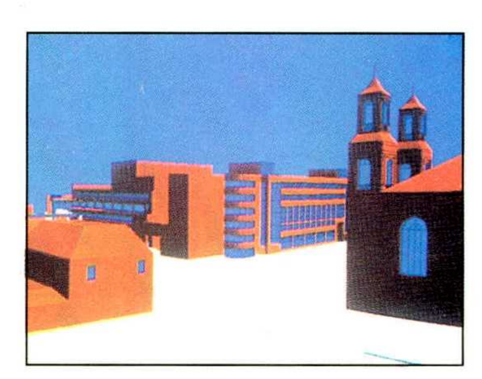
CSM

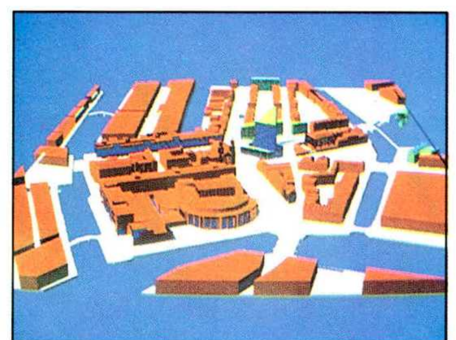


Die Aufnahmen entstanden in der Discothek 'Luggi' in Peißenberg.

AXIS

3D Zeichnungen & Animationen auf ST und PC







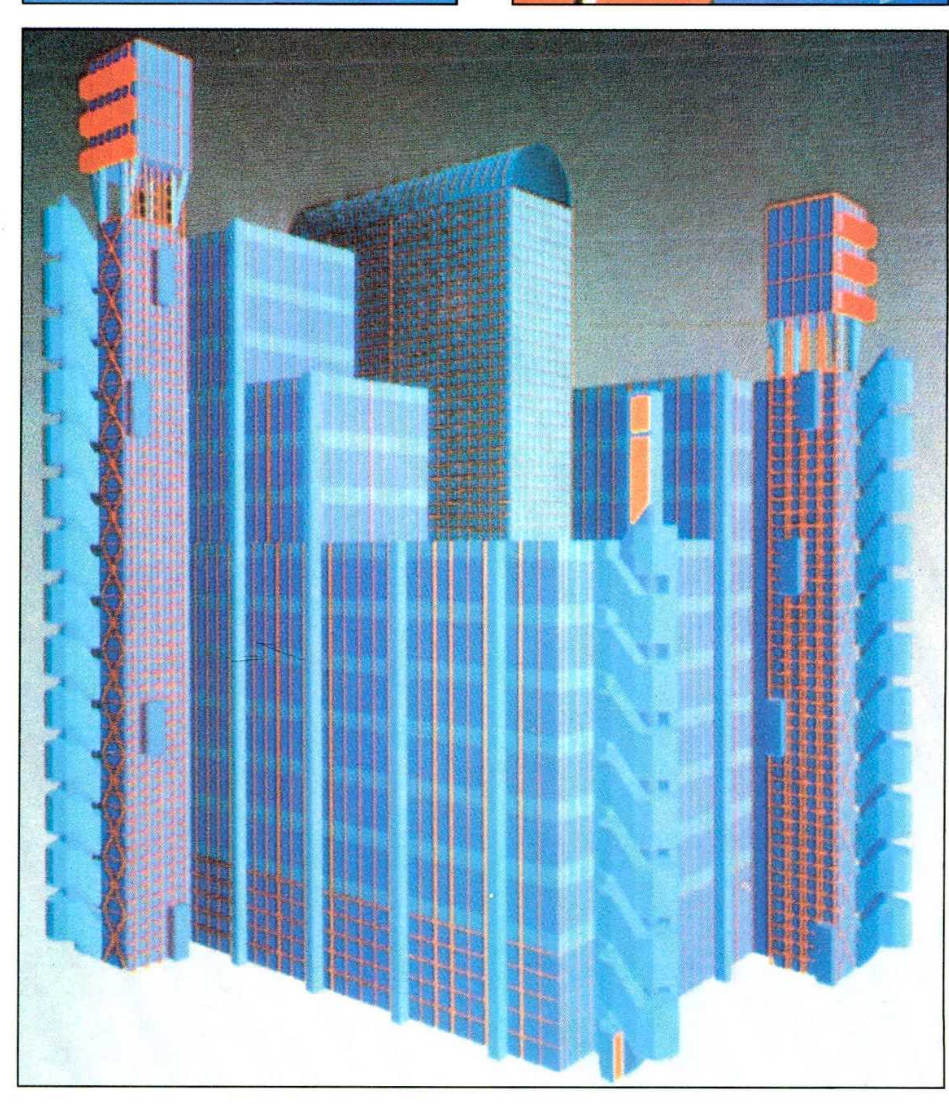
Erste Versionen von Axis waren bereits auf der ATARI-Messe in Düsseldorf im Jahre 1987 zu sehen. Endgültig vorgestellt wurde das Produkt des holländischen Herstellers Modern Medium aber erst auf der diesjährigen CeBit; eine Vorstellung, die durch eine auf Videokassette verfügbare und auch verhältnismäßig lange Animation recht eindrucksvoll war. Nun, Vorhang auf für Axis.

Axis ist ein maus-, nicht aber ein GEM-gesteuertes Produkt. Dies ist umso erstaunlicher, als Axis für die Grafikausgabe sowohl auf dem PC wie auf dem ST das GEM-Output-Programm verwendet. Die Bedieneroberfläche ist die eines typischen PC-Produktes:

Am linken Bildschirmrand befinden sich Menütitel, die man mit der Maus anklikken kann. Daraufhin erscheinen Untermenüs. Usw., bis man sich zu der gewünschten Option durchgehangelt hat. Das hört sich zwar in der Beschreibung ähnlich komfortabel an wie Pull-Down-Menüs, ist es aber leider nicht; erstens gibt es die Hauptmenüs in mehreren Modi, zweitens gibt es sehr viele Untermenüs, und drittens muß man jeden Menüpunkt explizit anklicken, draufzeigen genügt nicht.

Am oberen Bildschirmrand ist eine freie Fläche, die das Programm für Dialoge mit dem Benutzer verwendet, in der rechten oberen Ecke werden wichtige Statusinformationen angezeigt.

Auch für die Auswahl von Dateinamen wird keine Fileselektorbox verwendet; aber wenigstens kann man sich die vorhandenen Dateien anzeigen lassen (den Namen muß man dann aber trotzdem eintippen). Alles in allem also ein recht



ungewohntes Erscheinungsbild für ein ST-Programm. Davon sollte man sich jedoch zunächst nicht abschrecken lassen, denn für eine bestimmte Klasse von dreidimensionalen Objekten ist die Arbeitsweise von Axis recht komfortabel.

Konstruktion

Axis ist offensichtlich vor allem für die / von der häusererrichtende(n) Zunft konzipiert; die vorhandenen primitiven Objekte lassen doch sehr deutlich darauf schließen. Es gibt innerhalb von Axis nicht die üblichen Konstruktionswerkzeuge, wie man sie von Programmen wie CAD 3D oder Cyber Sculpt her kennt. Man kann also nicht eine zweidimensionale Figur oder Schablone per Rotation oder Translation in einen dreidimensionalen Körper verwandeln, jedenfalls nicht in diesem allgemeinen Sinn.

Die vier Grundelemente des Axis-Designs sind die horizontale (xy-)Ebene, die Mauer (die aus einem Grundriß-Polygon und einer Höhe gebildet wird), das Haus (einfaches Klötzchen mit Giebeldach) und der Zylinder (der recht flexibel ist - schiefe Türme sind kein Problem - ebensowenig Zylindersegmente oder Ausschnitte). Alle Grundformen werden aus Polygonen zusammengesetzt.

Zusätzlich gibt es noch zwei 'freie' Objektformen, die sich sehr ähnlich sind und sich eigentlich nur in der Art der Eingabe unterscheiden; Beide sind letztlich einfache im Raum positionierte Polygone, deren Eckpunkte per Tastatur eingegeben werden müssen. Die Axis-Grundelemente sind durch einige Parameter variierbar; das Haus baut auf einer viereckigen Grundfläche auf, wobei Haus- und Giebelhöhe frei wählbar sind. Die Eckpunkte für alle Grundobjekte können wahlweise per Maus (mit Snap-Funktionen) oder per Tastatur eingegeben werden. Zusatzparameter wie Höhe oder Farbe müssen immer den Weg über die Tastatur nehmen.

Jedes in Axis definierte Polygon, sei es nun einzeln definiert oder Teil eines Hauses oder Zylinders, kann mit sogenannten Details versehen werden. Das sind 'Strichzeichnungen', mit denen die Oberfläche quasi verziert werden kann. Auf eine Hauswand könnte man zum Beispiel Fenster und Türen malen.

Selbstverständlich enthält Axis auch Funktionen, mit denen man nachträglich einzelne Eckpunkte eines erzeugten Objektes edieren kann, Fehler sind jederzeit korrigierbar.

Um aus den doch recht beschränkten Grundformen komplexe Objekte erzeugen zu können, enthält Axis sogenannte Gruppenfunktionen. Hiermit kann ein Objekt oder eine Gruppe von Objekten kopiert, gespiegelt, rotiert und verschoben werden. Die meisten Funktionen normaler Konstruktionsprogramme, z.B. Rotationskörper, lassen sich so, bei eingeschränktem Komfort, simulieren. Andererseits lassen sich mit der Funktion sehr komfortabel komplexe Szenen aus Sequenzen von einfachen Objekten zusammensetzen. Im Handbuch sind dazu zwei Beispiele enthalten: Die sehr mühsame Erzeugung einer Kugel und die extrem einfache Konstruktion einer Wendeltreppe aus einzelnen Stufen. Das funktioniert so: Bei der Kopierfunktion (um ein Beispiel herauszugreifen) kann man angeben, wieviele Kopien erzeugt werden sollen. Jede Kopie kann aber um jede Achse rotiert oder verschoben werden. Die Wendeltreppe entsteht also sehr schnell, indem man eine Stufe konstruiert und diese dann ein paarmal kopiert, jeweils um einen bestimmten Winkel um den Treppenpfosten verdreht und entlang desselben verschoben. Die Gruppenfunktionen verdienen auf jeden Fall ein großes Lob.

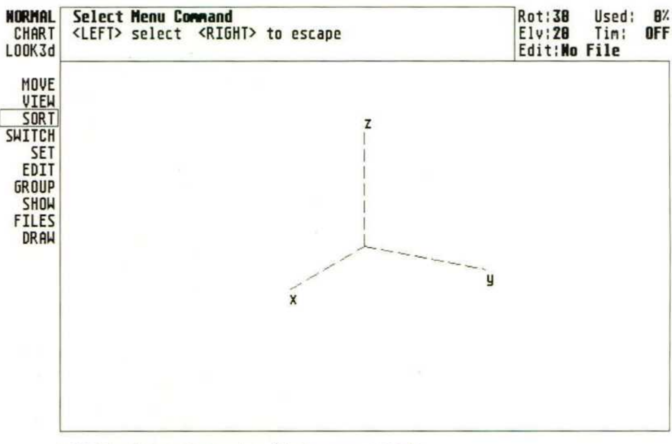
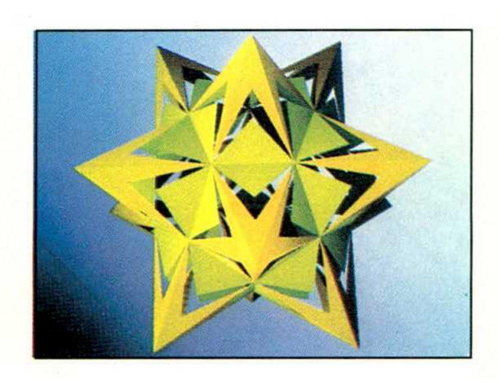
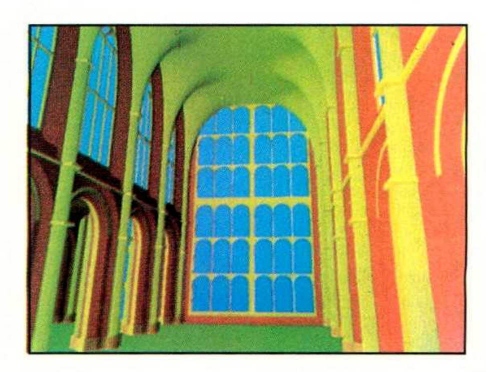


Bild 1: Die Arbeitsoberfläche von AXIS

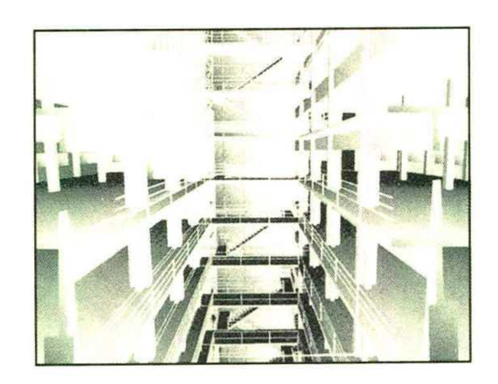
Durch die doch recht stark koordinatenbezogene Eingabetechnik wird es sehr schwierig und unintuitiv, Objekte in einer Richtung zu manipulieren, die nicht auf irgendeine einfache Weise an den Hauptkoordinatenachsen orientiert sind. Wie zahlreiche ergonomische Tests beweisen, ist das abstrakte räumliche Denken der wenigsten Menschen so beschaffen, daß Rotationen um eine beliebige Raumachse per Winkeleingabe der Teilrotationen um die drei Hauptachsen effizient möglich sind. Dies ist überhaupt das Hauptproblem im Design der Benutzerschnittstelle von Axis: Zwar ist die Bedieneroberfläche recht effizient, sie ist jedoch in weiten Teilen überhaupt nicht intuitiv, was aber gerade für 3D-Konstruktionsprogramme, die als echte Werkzeuge den Zeichentisch auch bei kreativen Prozessen ergänzen sollen, ganz besonders wichtig ist, was die oben bereits erwähnten Erkenntnisse über das räumliche Denken des Menschen belegen. Axis legt demjenigen, der in Koordinaten zu denken imstande ist, beileibe keine Hindernisse in den Weg; wer damit jedoch Schwierigkeiten hat (wohl die große Masse der Anwender), wird in der Bedienerführung von Axis wenig Hilfe finden.





Rendering

Axis stellt, wie auch CAD 3D, Objekte entweder als Drahtmodelle oder mit Hilfe der sogenannten 'Flat Shading' genannten Schattierungsmethode dar. 'Flat Shading' bedeutet, daß die Beleuchtung (und damit die Helligkeit) eines beliebigen Polygons einmal berechnet wird, und dann für das ganze Polygon konstant bleibt. Ein aus Polygonen zusammengesetzter Zylinder sieht dann eben aus wie ein aus Polygonen zusammengesetzter Zylinder. Auf einem PC ist dieser Ansatz auch vernünftig, weil alle Techniken, die einen aus Polygonen zusammengesetzten Zylinder bei der Darstellung rund aussehen lassen, erstens mit 16 Farben kaum sinnvoll einsetzbar und zweitens auch wegen ihres hohen Bedarfes an Rechenleistung kaum für PCs geeignet sind.



Aus dem vorigen Absatz folgert der aufmerksame Leser, daß es möglich ist, Lichtquellen zu positionieren, die die Szenerie ausleuchten.

Eine Darstellung mit 'Flat Shading' erfordert natürlich auch die Berechnung der verdeckten Flächen; nur diejenigen Flächen der Szene dürfen gezeichnet werden, die vom Standpunkt des Betrachters aus auch wirklich sichtbar sind. Da alle Verfahren hierfür recht aufwendig sind, kann man sich auch behelfen, indem man die Flächen zumindest in der richtigen Reihenfolge zeichnet, von hinten nach vorne nämlich. Dabei werden die hinten liegenden von weiter vorne liegenden übermalt, so daß nachher nur die richtigen Flächen zu sehen sind. Diesem Ansatz, bekannt unter dem Namen 'Painters Algorithm' (weil man wie ein Ölmaler ein Objekt über ein anderes malt), folgt, wie praktisch alle 3D-Programme auf Mikrocomputern, auch Axis. Dabei stehen zwei Qualitätsstufen zur Verfügung: 'Simple' und 'Extended'. Der erste Modus ist schnell, macht aber viele Fehler, während der zweite um einiges langsamer, aber auch entsprechend korrekter arbeitet. Ein allgemeines Problem des Algorithmus' ist seine Unfähigkeit, mit bestimmten Spezialfällen (z.B. zyklischer Überlappung von Polygonen) fertigzuwerden. Bei typischen aus den Axis-Primitiven zusammengesetzten Szenen sind die Ergebnisse aber sehr gut.

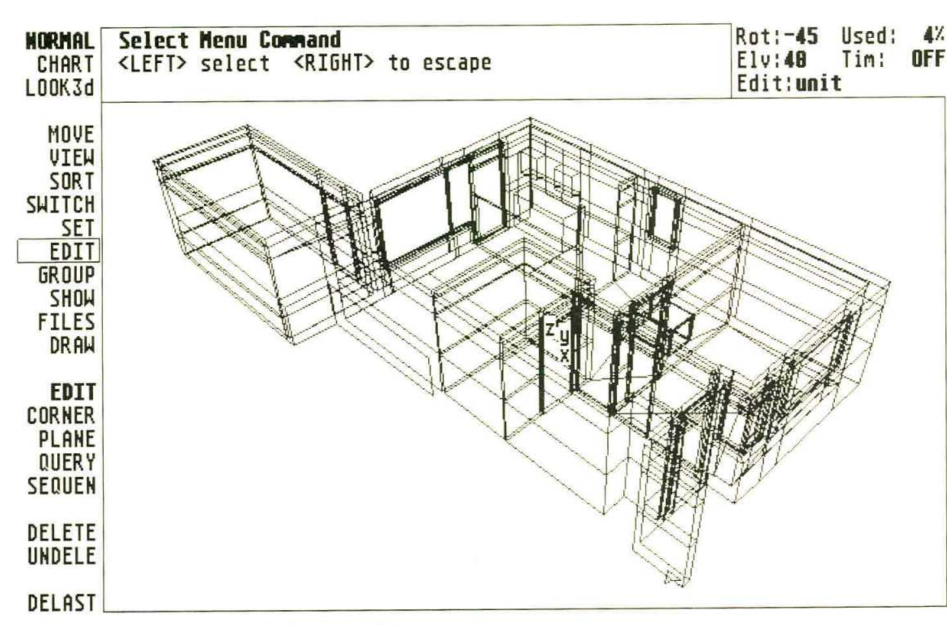


Bild 2: Ein Büro im Drahtmodell...

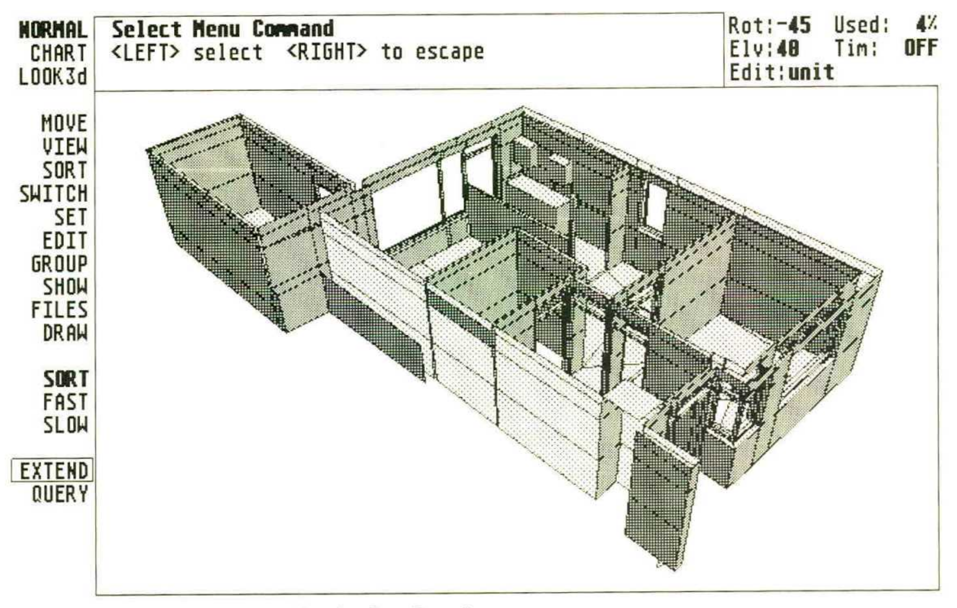


Bild 3: ... und mit Hidden-Surface-Berechnung

Natürlich sind sowohl perspektivische als auch parallele Projektionen möglich und Betrachterstand- und Blickpunkt sind selbstverständlich frei wählbar.

Eine besondere Zusatzfunktion verbessert die Resultate noch: Wenn man Objekte (z.B. Häuser) auf einer horizontalen Ebene positioniert, kann man den Schattenwurf dieser Häuser auf der Ebene für eine bestimmte Lichtposition berechnen lassen. Diese Schattenwürfe werden auf der Ebene behandelt wie Details und können auch entsprechend bearbeitet und gefärbt werden. Eine gute Idee.

Die Ergebnisse von Axis speziell für Architekturbilder sind wirklich erstaunlich, wenn man den geringen Kostenaufwand berücksichtigt, den das Programm verursacht. Im Demo-Video wird ein Flug über Rotterdam dargestellt, der auf einem PC berechnet wurde. Beachtlich.

Animation

Axis beherrscht noch eine weitere Dimension, nämlich die Zeit. Der Zustand eines Objektes (z.B. die Position) kann über die Zeit verändert werden. Genauso ist es auch möglich, die einzelnen Phasen eines Gruppenkommandos (z.B. Kopieren) als Animationsphasen zu verwenden und so etwa die Entstehung der vielgerühmten Wendeltreppe zu betrachten. Auch der Beobachter kann sich bewegen. Die Ergebnisse einer Animation können

als Bildfolge auf Diskette (bzw. besser auf Harddisk) abgespeichert und dann mit einem Abspielprogramm gezeigt werden. Die maximal mögliche Filmlänge hängt dabei natürlich vom verfügbaren Rechnerspeicher ab. Modern Medium bietet aber auch als Sonderservice die Übertragung von Bildfolgen auf Videobänder an.

Leider gibt es keinen Preview-Modus, aber das wäre von einem ST wohl auch etwas viel verlangt.

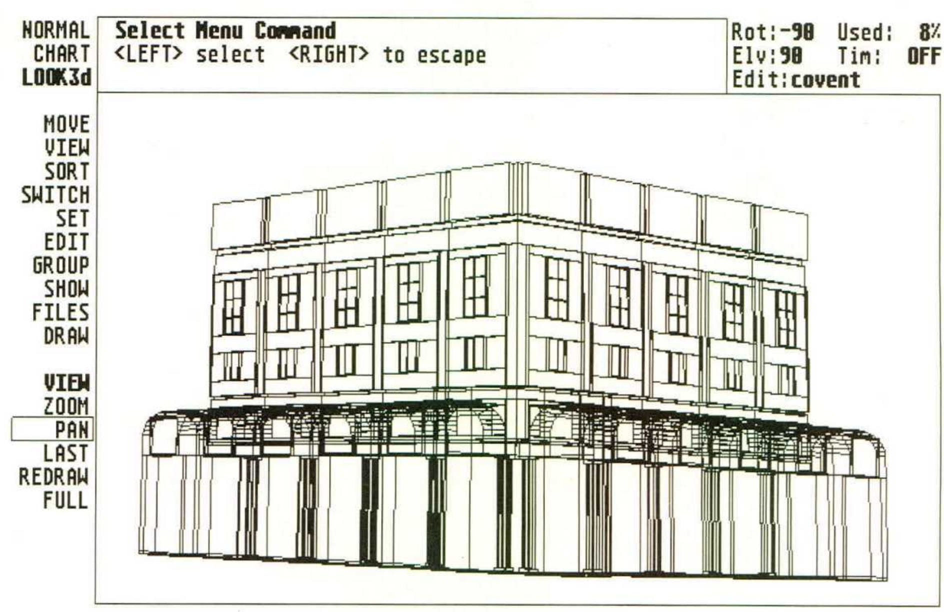


Bild 4: Ein Teil Londons, Covent Garden, im Drahtmodell...

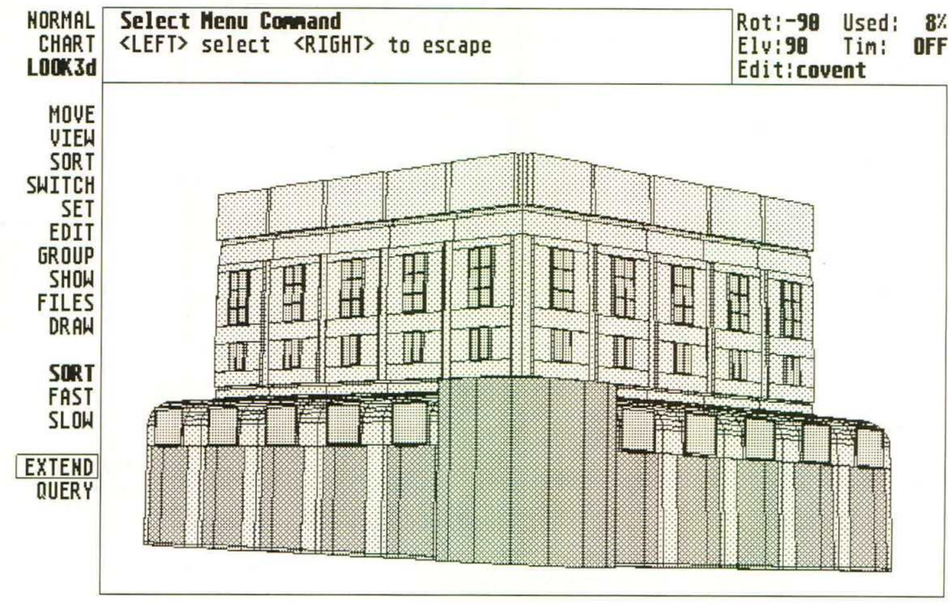


Bild 5: ... und mit Hidden-Surface-Berechnung

Dateien

Axis-ST kann mit seinen PC-Kollegen kommunizieren, ebenso, wichtig für Konstrukteure, mit AutoCAD. DXF-Dateien können sowohl für Objekte als auch als Details auf Objekten verwendet werden.

Handbuch

Das deutsche Handbuch zu Axis, das übrigens in englischer Sprache gehalten ist, kann man nur noch als katastrophal bezeichnen. Es handelt sich um eine Übersetzung aus dem Holländischen, die aber sprachlich unter aller Kritik ist. Man kann sich zwar hindurchwursteln, aber das Lesen ist eine Qual. Trotz der recht engen Sprachverwandschaft zwischen Holländisch und Deutsch: Eine schlechtere Übersetzung ist mir bisher nicht untergekommen. Auch inhaltlich könnte einiges besser erklärt sein, so z.B. der Animationsteil. Große Bitte an den deutschen Vertrieb: Eine Neufassung des Handbuches ist dringend nötig.

Ergänzungen

Außer dem bereits erwähnten Video-Service bietet der Hersteller auch Trainingskurse zu Axis an. Auch ein 'Konstruktionsservice' existiert, so daß Filme nach Vorlage als Dienstleistung berechnet werden können. Dabei können übrigens bis zu 350 000 Farben verwendet werden, was die Qualität des Ergebnisses im Vergleich zu ST-Ausgaben noch einmal erheblich verbessert.

Zusätzlich zu Axis und dem Abspielprogramm für Animationen enthält die ST-Version ein Programm, das die Verknüpfung einer Axis-Animation und eines Degas-Bildes erlaubt. Eine gute Idee.

Fazit

Axis ist ein sehr ambitioniertes Programm, das sich durch einen professionell orientierten Support auszeichnet. Ein 3D-Programm für den Einsatz in kreativen Berufen muß aber heute meines Erachtens mehr Intuitivität und Einfachheit in der Bedienung mitbringen. Selbst auf PCs gibt es in diesem Bereich bereits überzeugendere Programme. Die Funktionen selbst sind sehr leistungsfähig, besonders Architekturmodelle sollten sich mit den vorhandenen Primitiven meist in erträglichen Zeiten erzeugen lassen.

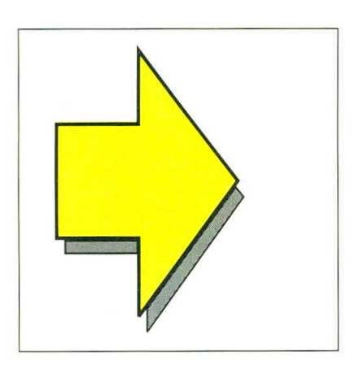
Dennoch, im direkten Vergleich mit der Cyber-Serie (Cad 3D, Cyber Sculpt, Cyber Paint, im Vertrieb M&T) kann Axis eigentlich nur die Pluspunkte Professionalität des Supports und Portabilität (besonders wegen der Ausgabemöglichkeit mit mehr Farben und höherer Auflösung) verbuchen. Der Preis für diese Pluspunkte ist allerdings als Extra zu bezahlen und seine Höhe ist uns unbekannt. Auch Architekturmodelle lassen sich wahrscheinlich in vielen Fällen leichter mit Axis erstellen, solange eben nicht Formen auftauchen, die erst mühsam über die Gruppenfunktionen erzeugt werden müssen. Das kreativere und vielseitigere Werkzeug für den ambitionierten Amateur-Animator ist jedoch eindeutig die Cyber-Serie.

Sämtliche Farbbilder sind auf einem PC mit einer 256 Farbenkarte berechnet.

CS

Bezugsadresse:

Modern Medium Deutschland
Hans Richter
Hagener Str. 65
5820 Gevelsberg
Tel.: 02332/2706



INTERLINK ST

INTERLINK ST ist das komfortabelste DFÜ-Programm für den ATARI ST und damit ideal für den Einsteiger und den Profi. So urteilen zumindest die Besitzer, die die Kommunikation und den weltweiten Datenaustausch mit Hilfe von INTERLINK ST nicht mehr missen möchten. Wann gehen Sie auf die Datenreise?

INTERLINK ST -
DFÜ im Griff
Unverbindliche Preisempfehlung DM 79,-

G+PLUS NEU!

G+Plus ist ein vollständiger Ersatz für GDOS. Mit G+Plus wird GEM nicht mehr gebremst! Fonts und Gerätetreiber werden ohne Neustart des Rechners bei Aufruf eines Programms automatisch nachgeladen! Im Gegensatz zu GDOS braucht G+Plus nicht entfernt zu werden, um die Betriebssicherheit von anderen Programmen zu gewährleisten.

G+PLUS -
Der GDOS-Ersatz
Unverbindliche Preisempfehlung DM 79,-

HOTWIRE NEU!

Die Shell für den geplagten ST-Anwender. Laufwerke, Ordner öffnen, Namen anklikken - Das alles gehört der Vergangenheit an. HotWire startet jedes ST-Programm durch Drücken einer Taste, egal wo das Programm steht! Hotwire erlaubt dazu über 400 Tastenkombinationen. Dazu Autostart bei Einschalten des Rechners und vieles mehr.

HOTWIRE -
Die Starthilfe
Unverbindliche Preisempfehlung DM 79,-

MIDIMAX NEU!

MIDIMAX ist das Utility für den MIDI-Anwender. MIDI-Macros, die von jedem MIDI-Ereignis getriggert werden können; Realtime Multivoice, Multichannel Modal Harmonisierung. Akkorde mit bis zu 18 Noten sowie die Möglichkeit das Keyboard zu splitten, maximal in jede Taste! Der ATARI ST als intelligente MIDI-Thru-Box.

MIDIMAX -
Das MIDI-Utility
Unverbindliche Preisempfehlung DM 129,-

Vertrieb in d. Schweiz:	Vertrieb in Österreich:
DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07 (ausg. Turbo ST)	Dipl.-Ing. R. Temmel Ges.m.b.H. & Co.KG A-5440 Golling Markt 109 Tel.: 06244/7081-17 Fax.: 06244/7188-3
Alle Programme und Handbücher in Deutsch. Nur mit Einsendung der Registrierkarte direkt an BELA Computer können Sie den Update-Service in Anspruch	Vertrieb in Frankreich: AROBACE 2, rue Piémontési F-75018 Paris Tel.: 1/42235044 Fax.: 1/42545631

MULTIDESK NEU!

Multidesk lädt bis zu 32 .ACC-Programme nach und verbraucht nur einen Eintrag im DESK-Menü. In sich selbst geladen beschränkt nur noch der max. verfügbare Speicher die Anzahl der Accessories. Das Nachladen ist jederzeit möglich! Oder starten Sie jedes Accessory wie ein normales Programm.

MULTIDESK -
Der ACC-Manager
Unverbindliche Preisempfehlung DM 79,-



REVOLVER

Wo andere Programme den Dienst quittieren, da bietet REVOLVER Sicherheit. Resetfest in jedem Rechnerteil und mit umfangreiche Utilityfunktionen ist REVOLVER ideal für Programmierer, Musiker und Anwender, die mehr aus ihrem ATARI ST machen wollen.

REVOLVER -
Der Profi-Switcher
Unverbindliche Preisempfehlung DM 129,-

STOP NEU!

Einbruch und Datendiebstahl - kein Thema auf dem ST? Mit STop schützen Sie persönliche Daten, Programme oder Artikel- und Kundendateien vor fremden Zugriff. Nur über die Paßwörter (maximal 64 Stellen) ist der Echtzeitzugriff auf die vollständig kodierten Daten möglich.

STOP -
Der Datentresor
Unverbindliche Preisempfehlung DM 129,-

TURBO ST V1.6!

Die Lösung: Laden statt Löten! Machen Sie Ihrem ATARI ST "Beine" und beschleunigen Sie die Anzeige von Texten auf dem Bildschirm auch ohne Blitterchip. Turbo ST ist sogar noch schneller als der Blitter und damit besonders interessant für Besitzer der STs, in denen der Blitter überhaupt nicht vorgesehen ist.

TURBO ST -
Der Softwareblitter
Unverbindliche Preisempfehlung DM 79,-



BELA Computer Layout- und Vertriebs GmbH Unterortstr. 23-25 D-6236 Eschborn Tel: 06196/481944 Fax: 481930

"Vor die Wissenschaft haben die Götter die Statistik gestellt", mag so mancher wissenschaftlich tätige Zeitgenosse klagen, und nicht wenige kämpfen sich tapfer durch Zahlenberge, Meßwertreihen und ähnliche Gebilde. Mußte der so geplagte Zeitgenosse vor nicht allzu langer Zeit noch lochkartenstanzend seine Daten in den Computer eingeben, so verfügt er heute über flexible und einfach zu handhabende Statistikpakete auf PC-Basis.



Wissenschaftliche Statistik auf dem ATARI ST

Abgesehen von Statistikpaketen, die unter dem Betriebssystem MS-DOS laufen und nicht selten die Eintausend-Mark-Grenze weit überschreiten, wurden für den ATARI ST in den letzten 2 Jahren einige Statistikpakete entwickelt bzw. adaptiert, die vom Preis-/Leistungsverhältnis her äußerst attraktiv sind.

Im folgenden werden vier dieser Pakete besprochen werden, wobei drei dieser Produkte (ST-Statistik, Variana, WiStat/ Regression ST) eigenständige Entwicklungen sind; das vierte Produkt, die "Statistik Library" von OMIKRON, ist eine Statistikbibliothek auf der Basis des OMIKRON.BASICs und setzt den diesbezüglichen Interpreter voraus.

Bei der Besprechung der einzelnen Verfahren habe ich versucht, die sachlogische Stufenfolge bei der statistischen Datenverarbeitung mit Hilfe eines Computers einzuhalten (siehe Tabelle 1):

Zuerst müssen die Daten irgendwie in den Rechner gelangen (*Daten-Eingabe*). Fertige Dateien müssen manchmal verändert bzw. die in ihnen enthaltenen Daten vor der statistischen Analyse aufbereitet werden (*Datei-Manipulation, Daten-Aufbereitung*). Nun endlich kann sich der statistische Eifer frei entfalten, und es geht an die Berechnung grundlegender statistischer Maßzahlen (*Grund-Statistiken*). Damit nicht zufrieden, kann man sich als nächstes auf die Analyse grundlegender Zusammenhänge stürzen (*Korrelation / Kontingenz, Regression*). Ist man bis dahin immer noch nicht frustiert, geht es auf zum Vergleich zweier oder mehrerer Stichproben (unter "Stichprobe" kann man - für alle praktischen Zwecke - eine bestimmte Menge erhobener oder sonstwie produzierter Daten bzw. Meßwerte verstehen).

Doch halt! Einige dieser Vergleichstests setzen voraus, daß wir über die Stichprobe(n) einige Vorinformationen besitzen. So sollte man bestimmte Tests nur dann verwenden, wenn die Daten, die man dazu heranzieht, "normalverteilt" sind, die Verteilung also annähernd die bekannte "Glockenform" hat (ob dies so ist, kann man natürlich ebenfalls statistisch testen).

Verzichtet man auf derartige Vorinformationen wie z.B. Normalverteilung der Daten u.a., so kann man sich zum Vergleich von Stichproben der sog. *non-parametrischen Tests* bedienen. Legt man aber derartige Vorinformationen

(Voraussetzungen) zugrunde, lassen sich die statistischen Vergleiche mit *parametrischen Tests* durchführen. Will man dabei nur zwei Stichproben miteinander vergleichen, reichen *einfache parametrische Tests*, bei drei und mehr Stichproben ist man auf komplexere parametrische Tests wie z.B. *Varianzanalysen* angewiesen.

Unter *weiteren Verfahren* sind eine Reihe z.T. sehr komplexer statistischer Prozeduren aufgeführt, die dem fortgeschrittenen Statistiker vertraut sein dürften.

Genug des Testens, her mit den Ergebnissen! Aber wie und wohin? Bei der *alphanumerischen Ausgabe* sind mehrere Möglichkeiten denkbar (siehe Tabelle 1). Dies gilt auch für die *graphische Ausgabe* von Ergebnissen: hier ist neben dem Ausgabemedium [Bildschirm, Drucker, Disk(ette)] vor allem auch die Ausgabeform (Punkte-, Liniendiagramm etc.) interessant; weiterhin ist wünschenswert, daß sich die Graphiken mittels eines eigenen Grafik-Editors direkt verändern lassen.

Schließlich interessiert viele Anwender noch die Frage, ob sich die ausgegebenen Daten bzw. Graphiken in andere Programme übernehmen lassen, z.B. in ein Textverarbeitungsprogramm. Die Druckausgabe ist übrigens bei den hier vorgestellten Programmen im Prinzip auf EPSON- bzw. NEC-kompatiblen Matrix- bzw. Laserdruckern möglich, z.T. sind durch die Distributoren spezifische Druckertreiber erhältlich. Doch nun zur Besprechung der einzelnen Programme:

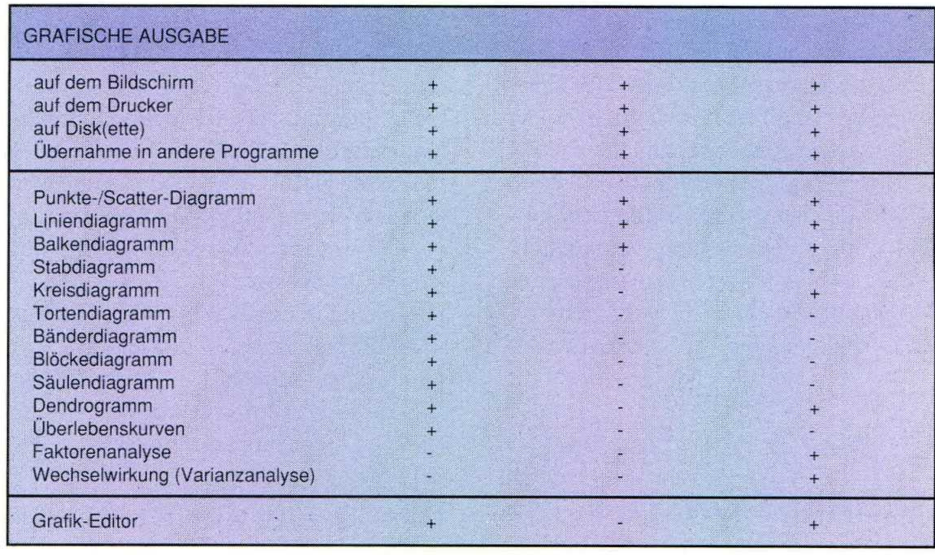
ST-STATISTIK 2.6

Bei der Dateneingabe sind dem Anwender von ST-Statistik nur durch den verfügbaren Massenspeicher Grenzen gesetzt, d.h. gegenüber anderen Programmen sowie früheren Versionen dieses Programms können mit ST-Statistik sehr große Datenmengen verarbeitet werden. Die Daten können sowohl mit einem eigenen Editor als auch durch Übernahme aus anderen Programmen (VIP, LOGISTIX, ADIMENS, 1st Word plus, etc.) eingegeben werden. Dabei können Fremddateien in ASCII-Form durch das mitgelieferte "STATCON.PRG" in Binärdateien konvertiert werden. Bei der Dateimanipulation und Datenaufbereitung fällt besonders positiv der "Datenfilter" ins Auge, mit dessen Hilfe fast beliebig Teilmengen aus der Ursprungsdatei herausgefiltert werden können.

	ST-STATISTIK	VARIANA	WISTAT/REGR.ST
DATENEINGABE			
eigener Editor	+	+	+
ASCII-Dateien	+	+	+
max. Dateigröße	begrenzt durch Massenspeicher	begrenzt (s.Text)	begrenzt durch RAM
DATEIMANIPULATION			
Datei vergrößern	+	+	+
Datei verkleinern	+	+	+
Datei teilen	+	-	+
Dateien zs.fügen	+	+	+
Matrix transponieren	+	-	+
DATENAUFBEREITUNG			
Daten filtern	+	-	-
Daten sortieren	+	+	+
Daten transformieren	+	+	+
Differenzen bilden	-	+	-
GRUNDSTATISTIKEN			
arithm. Mittelwert	+	+	+
Standardabweichung	+	+	+
Median, Zentilwerte	+	(+)	+
Schiefe, Exzeß	+	(+)	+
Häufigkeitstabelle	+	+	+
KORRELATION / KONTINGENZ			
Produkt-Moment-K.	+	+	+
Rang-Korrelation	+	+	+
Kontingenz	+	(+)	+
REGRESSION			
lineare Regression	+	+	+
nicht-lineare Regress.	+	-	+
multiple Regression	+	-	+
NON-PARAMETRISCHE TESTS			
Chi-Quadrat-Test	+	+	+
Kolmogorov-Smirnov-Test	-	+	+
Wilcoxon-Test	+	+	+
U-Test (Mann-Whitney)	+	+	+
H-Test (Kruskal-Wallis)	+	+	+
Median-Test	-	+	-
Friedman-Test	+	+	+
PARAMETRISCHE TESTS			
einfache parametr. Tests:			
- für unabh. Stichproben:			
F-Test	+	+	+
t-Test (Student)	+	+	+
t-Test (Welch)	+	+	+
- für abhäng. Stichproben:			
t-Test (korr.)	+	+	+
t-Test (Ferguson)	+	+	-
Varianzanalysen:			
einfaktorielle VA	+	+	+
zweifaktorielle VA	+	+	+
dreifaktorielle VA	-	-	+
multivariate VA	-	-	+
WEITERE VERFAHREN			
Konfigurations-frequenzanalyse	+	-	-
Clusteranalyse	+	-	+
Faktorenanalyse	+	-	+
Diskriminanzanalyse	-	-	+
Itemanalyse	+	-	+
Kaplan-Meier-Test	+	-	-
Mantel-Haenszel-Test	+	-	-
ALPHANUMERISCHE AUSGABE			
auf dem Bildschirm	+	+	+
auf dem Drucker	+	+	+
auf Disk(ette)	+	+	+
Übernahme in andere Programme	+	(+)	+

Die statistischen Verfahren decken nahezu den gesamten Bereich der gängigen und meistverwendeten Prozeduren ab. Lediglich bei den "Varianzanalysen" wäre eine baldige Erweiterung auf mehrfaktorielle Verfahren wünschenswert; die "Diskriminanzanalyse" wird man nur gelegentlich vermissen.

Dafür aber sind "Cluster-, Faktoren- und Itemanalysen" enthalten, und als besonderer Leckerbissen - und in keinem anderen der hier vorgestellten Programmpakete enthalten - bietet ST-Statistik die "Konfigurationsfrequenzanalyse" sowie Verfahren zur Schätzung bzw. zum Vergleich von "Überlebenskurven" ("Kaplan-Meier-Test, Mantel-Haenszel-Test").



GRAFISCHE AUSGABE			
auf dem Bildschirm	+	+	+
auf dem Drucker	+	+	+
auf Disk(ette)	+	+	+
Übernahme in andere Programme	+	+	+
Punkte-/Scatter-Diagramm	+	+	+
Liniendiagramm	+	+	+
Balkendiagramm	+	+	+
Stabdiagramm	+	-	-
Kreisdiagramm	+	-	+
Tortendiagramm	+	-	-
Bänderdiagramm	+	-	-
Blöckediagramm	+	-	-
Säulendiagramm	+	-	-
Dendrogramm	+	-	+
Überlebenskurven	+	-	-
Faktorenanalyse	-	-	+
Wechselwirkung (Varianzanalyse)	-	-	+
Grafik-Editor	+	-	+

Tabelle 1: Übersicht über die Funktionen der getesteten Programme

Die alphanumerische Ausgabe läßt nichts zu wünschen übrig, die Ergebnisse können wohlgeordnet in andere Programme, v.a. Textprogramme, übernommen werden.

Die graphische Ausgabe von ST-Statistik ist wahrlich beeindruckend und enthält nahezu alles, was diesbezüglich auf diesem Sektor realisiert werden kann. Ob man Ergebnisse im zwei- oder im dreidimensionalen Raum darstellen will, ein paar Tastendrucke bzw. Mausklicks, und schon steht die Grafik (siehe Abb. 1 und 2). Auch hier ein besonderer Leckerbissen: Bei der Clusteranalyse besteht die Möglichkeit, ein Dendrogramm zu erzeugen, was sogar ausschnittsweise vergrößert (per "Zoomen") werden kann (siehe Abb. 3).

VARIANA

Nach dem Programmstart und dem kurzfristigen Erscheinen eines Logos gibt Variana ein Arbeitsfeld frei, das in zwei Bereiche und eine Menüzeile aufgeteilt ist. In der linken Bildhälfte (Bereich 1) können Dateien angezeigt und ediert werden, in der rechten (Bereich 2) befinden sich anwählbare Edierfunktionen. Bei der Arbeit mit dem Edierfenster (Aufbau nach GEM) ist zu beachten, daß manche Grafiksymbole (z.B. Maximalgrößenfeld) andere Funktionen als sonst bei GEM üblich besitzen.

Apropos Fenster: positiv fällt hier bei Variana auf, daß maximal 6 Fenster (einschließlich Edierfenster) geöffnet werden können.

Grundsätzlich lassen sich bei Variana zwei Dateien gleichzeitig laden, wobei eine Datei (Vordergrunddatei) direkt bearbeitet werden kann, die andere (Hintergrunddatei)im RAM-Speicher steht.

Bei der Dateneingabe sind bei Variana maximal 8000 Werte pro Stichprobe zulässig, die Anzahl der Stichproben wiederum ist ausschließlich durch den verfügbaren RAM-Speicherplatz begrenzt.

Die Dateimanipulation sowie die Datenaufbereitung ist auf mannigfache Weise möglich (siehe Tabelle 1), erfreulich ist hier die Möglichkeit der Bildung von "Differenzenstichproben" (Stichprobe X - Stichprobe Y).

Die verfügbaren Grundstatistiken schließen zwar solche ein, die in den anderen Programmpaketen z.T. nicht direkt berechnet werden (z.B. geometrisches und harmonisches Mittel), auf der anderen Seite fehlen wichtige Kenngrößen wie z.B. die direkte Angabe des Medians bzw. der Zentilwerte, die man bei Variana nur indirekt ermittelt (z.B. unter Bezug auf Summenkurven). Bei den statistischen Verfahren bietet Variana viele Prozeduren für einfache und gängige statistische Analysen (siehe Tabelle 1).

Bei den komplexeren Verfahren bietet es jedoch gegenüber den anderen Pro-

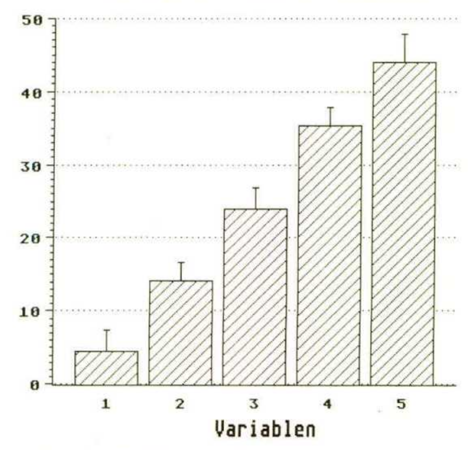


Abb. 1: ST-STATISTIK - Mittelwerte mit Fehlerbalken

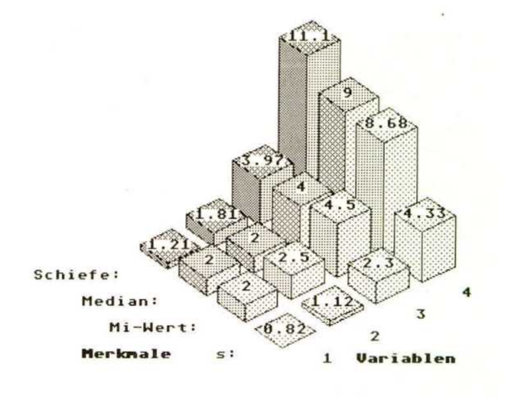


Abb. 2: ST-STATISTIK - Säulendiagramm

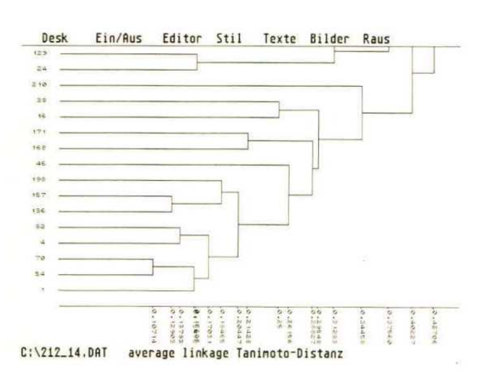


Abb. 3: ST-STATISTIK - Dendrogramm-Teilausschnitt durch "Zoomen"

grammpaketen deutlich weniger. Die alphanumerische und graphische Ausgabe ist, was die Ausgabeform angeht, eingeschränkt (siehe Tabelle 1); umfangreichere grafische Möglichkeiten fehlen, ebenfalls ein programmeigener Grafikeditor (Übernahme in andere Grafikprogramme und dortige Bearbeitung ist möglich). Positiv ist hier die bereits zuvor erwähnte Möglichkeit, mehrere Fenster gleichzeitig zu öffnen und in variabler Größe darstellen zu können (siehe Abb. 4).

WISTAT 2.0 / GRAPH REGRESSION ST 3.0

"WiSTat" ist ein Statistikprogramm, das für sich genommen keine graphische Ausgabe erlaubt, jedoch die Ergebnisse u.a. in Dateien ablegt, die durch das Programm "WiSTat Graph" aufrufbar sind, wobei dann die Ergebnisse graphisch dargestellt und ediert werden können. Auch zu "Regression ST" ist eine Schnittstelle vorhanden, wobei dieses Programm besonders für die Berechnung und graphische Darstellung von Regressionsanalysen entwickelt wurde, aber auch die Berechnung einiger Grundstatistiken erlaubt. Sowohl WiSTat als auch Regression ST besitzen eigene Editoren. Da WiSTat Graph und Regression ST vom selben Autor entwickelt wurden, nimmt es nicht wunder, daß viele in Regression ST realisierte Grafikmöglichkeiten auch in WiSTat Graph auftauchen. Für den Anwender bedeutet dies, daß die in Regression ST realisierten Möglichkeiten weitestgehend in der Kombination "WiSTat und WiSTat Graph" enthalten sind. Aus praktischen Gründen werden im folgenden diese drei Programme deshalb zusammen betrachtet (siehe dazu auch Tabelle 1).

Die Dateneingabe kann manuell erfolgen oder durch das Einlesen von Files, wobei grundsätzlich alle als ASCII-Dateien abgelegten Daten mit WiSTat eingelesen werden können, auch Dateien von 1st Word plus (ohne WP-Modus), Adimens ST, VIP Professional und natürlich Regression ST. Dabei werden durch die Funktion "Reinigen" Fremddateien von Buchstaben, Kommata etc. gereinigt.

Bei der Dateimanipulation und Datenaufbereitung kann u.a. eine Datenmatrix transponiert werden (Vertauschen der Zeilen mit den Spalten und umgekehrt), eine automatische Datenfilterung nach Zielkriterien ist jedoch nicht möglich. WiSTat erlaubt die Berechnung der gängigen Grundstatistiken; die Berechnung von Korrelationen und Regressionen (insbesondere unter Berücksichtigung von Regression ST) läßt kaum einen Wunsch offen.

Bei den non-parametrischen Verfahren sind die wesentlichen Prozeduren enthalten, bei den parametrischen Verfahren zeigt WiSTat sich vor allem bei den Varianzanalysen von seiner besten Seite; hier gehen die Möglichkeiten gar bis zur einfachen multivariaten Varianzanalyse. Positiv fällt hier auch auf, daß die Wahrscheinlichkeitswerte für die Richtigkeit der jeweiligen Nullhypothese als numerischer Wert und nicht nur als über oder unter der jeweiligen Signifikanzgrenze gelegen (sog. Sternchen- oder Doppelsternchenphilosophie) angegeben werden.

Bei den weiteren, komplexen Verfahren ist ebenfalls Erfreuliches zu vermelden. Zwar fehlen Verfahren wie die Konfigurationsfrequenzanalyse und die Berechnung von Überlebenskurven, dafür jedoch bietet WiSTat die Item-, Cluster-

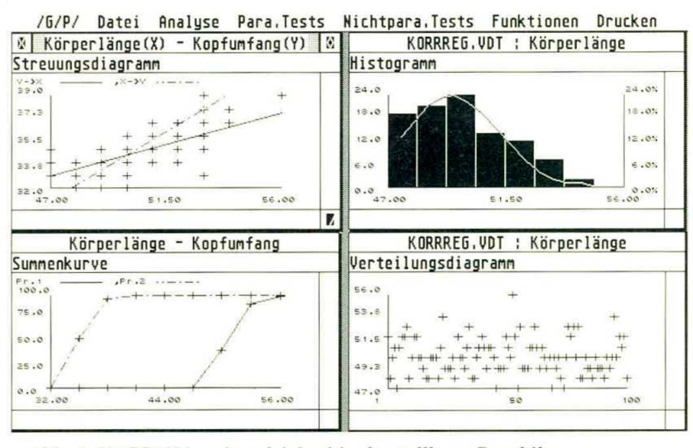


Abb. 4: VARIANA - vier gleichzeitig darstellbare Graphiken

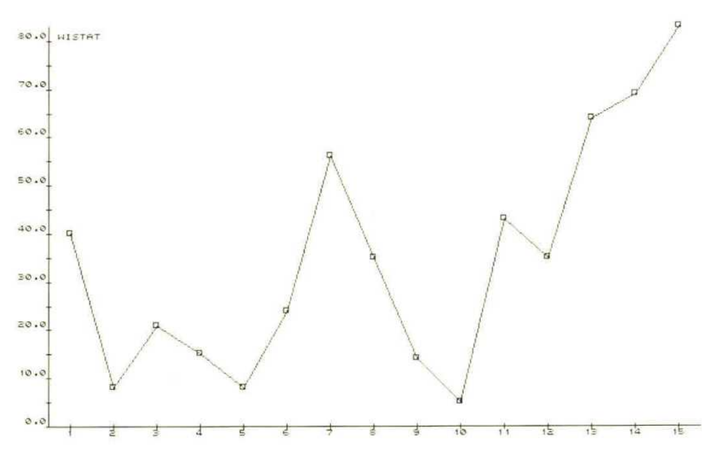


Abb. 5: WISTAT GRAPH - Liniendiagramm

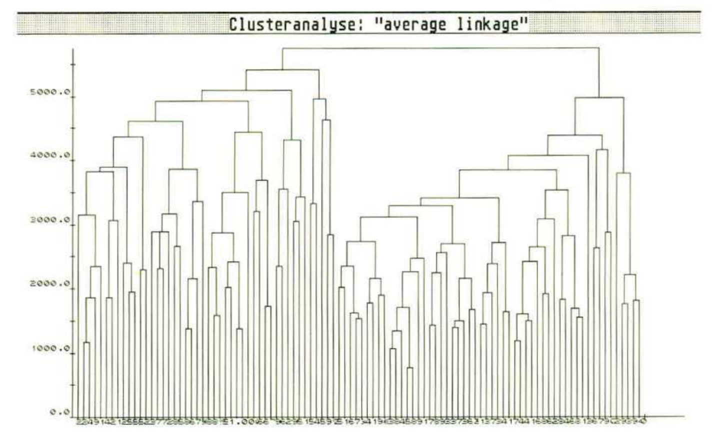


Abb. 6: WISTAT GRAPH - Dendrogramm

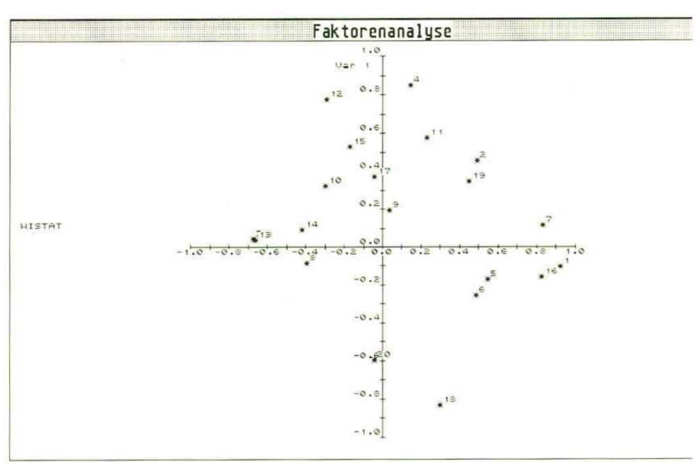


Abb. 7: WISTAT GRAPH - Faktorenanalyse

OMIKRON. **Basic**

NEU 3.0

ATARI ST
Schneider/Steinmeier
PROGRAMMIEREN IN OMIKRON BASIC
Heim-Verlag

ATARI ST
Schneider/Steinmeier
kurz & klar OMIKRON BASIC
Ein fundiertes Nachschlagewerk
Heim-Verlag

DIE SPITZEN-BÜCHER ZUM NEUEN ATARI-BASIC



MERKMALE:
▶ Omikron-BASIC 3.0 – das neue Standard-BASIC für den ATARI ST – wird immer häufiger zur Erstellung von großen u. kleinen Programmen eingesetzt. ▶ Das Buch „Programmieren in Omikron-BASIC" zeigt an vielen Beispielen die Entwicklung von Programmen. Anhand zahlreicher dokumentierter Listings wird der Umgang mit den besonderen Elementen dieser BASIC-Version erläutert. ▶ In einem 1. Abschnitt werden wichtige Unter- u. Hilfsprogramme vorgestellt: Druckeranpassung, universelle Zahlensystemumwandlung, Feststellen der Existenz einer Diskettendatei, Programmierung grafischer Blockoperationen mit dem BITBLT-Befehl u.a.m. ▶ Ein weiteres Kapitel behandelt den Einsatz von GEM-Funktionen in Omikron-BASIC-Programmen: Der Umgang mit Fenstern, Dialogboxen, der Fileselectbox u.a.m. Auch die Verwendung der zum BASIC gehörenden GEMLIB wird behandelt. ▶ Um die Programmierung zu erleichtern, werden einige Hilfsprogramme gezeigt, die z. B. das Kopieren mehrerer Files vereinfachen oder eine Datei in DATA-Zeilen umwandeln. Auch ein Editor für Mauszeiger u. einige Füllmuster fehlen nicht. ▶ Die grafischen Fähigkeiten des Computers u. die entsprechenden Anweisungen des BASIC-Interpreters werden anhand der Entwicklung einer dreidimensionalen Grafik mit Rotation u. Projektion sowie einer Turtle-Grafik u. weiteren kleinen Beispielen demonstriert. ▶ 4 größere Anwendungen bieten die Gelegenheit, sich mit umfangreicheren Projekten vertraut zu machen. ▶ Für Schüler u. interessierte Laien sind einige Anwendungen aus dem mathematischen Bereich gedacht. ▶ Zur Abrundung u. Auflockerung enthält das Buch einige Spielprogramme.

INHALT:
▶ Druckeranpassung ▶ Feststellen der Existenz einer Datei ▶ Das Verwenden mehrerer Bildschirme ▶ Einbinden von Funktionen in laufende Programme ▶ GEM-Programmierung ▶ Arbeiten mit Fenstern, Dialogboxen u. Menüzeilen ▶ Punktgenaue Eingabe auf dem Bildschirm ▶ Komfortable Filenameneingabe mit Fileselect ▶ Programmierung der Alertboxen ▶ Hilfsprogramme ▶ Editor für Mauszeiger u. Füllmuster ▶ Kopieren von mehreren Files ▶ Dreidimensionale Darstellung von Funktionen ▶ Turtle-Grafik ▶ Adreßverwaltung ▶ Ermitteln von Mittelwerten, Varianz u. Standardabweichungen ▶ Integration nach Simpson ▶ Spiele

ca. 400 Seiten — **DM 49,-** Best.-Nr. B-411

Diskette mit allen abgedruckten Programmen — **DM 39,-** Best.-Nr. D-431

INHALT:
ATARI hat sich entschlossen, endlich eine wirklich leistungsfähige Programmiersprache mit den Rechnern der ST-Serie auszuliefern. Daß die Wahl gerade auf Omikron-BASIC fiel, ist kein Wunder, denn diese Sprache ist nicht nur besonders einfach zu erlernen, sondern stellt zugleich einen Leistungsumfang zur Verfügung, der selbst das Schreiben professioneller Anwendungen erlaubt. Um die über 200 Befehle mit ihren zahlreichen Parametern nutzen zu können, ist eine alphabetische Übersicht der Kommandos und ihrer Möglichkeiten unverzichtbar, und genau dies liefert „Kurz & Klar Omikron-BASIC 3.0".

Sie erfahren, mit welchen Anweisungen man Matrizen multipliziert, invertiert oder eine Determinante ermittelt, wie man Linien und Kreise zeichnet oder mit welchem Befehl man eine Alertbox auf den Bildschirm bringt. Außer der reinen Befehlsübersicht, die an sich schon eine enorme Hilfe bei der Programmentwicklung darstellt, geben die zahlreichen Anhänge Auskunft über die Bedeutung der Modi des BITBLT-Befehls, die ASCII-Zeichen des ST's oder die Zuordnung der Tasten zu den Scan-Codes der INKEY$-Funktion. Auch die VT52-Codes werden in einem Anhang behandelt.

Wenn man einen Befehl aus einem bestimmten Anwendungsgebiet sucht, hilft der nach Anwendungen geordnete Befehlsindex weiter. Dem Besitzer eines Compilers, der für das Erstellen eigenständiger Programme erforderlich ist, hilft ein Abshcnitt über die Compilerdirektiven und andere Besonderheiten weiter.

Die neueste Auflage dieses praktischen Hardcover-gebundenen Nachschlagewerkes berücksichtigt selbstverständlich die neue Version 3.0 des Omikron-BASICs.

Über 200 Seiten — **DM 29,-** Best.-Nr. B-412

MERKMALE:
▶ Omikron-BASIC 3.0 ist der neue BASIC-Standard für den ATARI ST. Das vielfach bewährte „große Omikron-BASIC 3.0-Buch" gibt es nun in einer neuen Auflage, die alle Neuheiten berücksichtigt. ▶ Das Buch stellt einen leicht verständlichen Einstieg für den Anfänger in die Programmierung einer der leistungsfähigsten Sprachen für den ATARI ST dar. ▶ Jeder, der sich ernsthaft mit den Möglichkeiten dieser Programmiersprache beschäftigen möchte, benötigt dieses fundierte Lehrbuch. Aber auch der Umsteiger von einem anderen BASIC-Dialekt findet hier alle notwendigen Informationen, um mit Omikron-BASIC optimal arbeiten zu können. ▶ Dem Einsteiger bietet „Das große Omikron-BASIC 3.0-Buch" eine systematische und leicht verständliche Einführung, die von den Schleifen- und Programmstrukturen über die unterschiedlichen Variablentypen und die Arbeit mit Feldern bis zu den numerischen und Stringfunktionen reicht. Weiterhin findet der Leser Hinweise zum Umgang mit Unterprogrammen und Prozeduren. Aufbauend auf diesen Teil ist die Programmierung der Multitasking-Fähigkeiten anhand eines Drucker-Spoolers sowie der Einsatz von die Arbeit erleichternden abstrakten Datentypen erklärt. Weitere Abschnitte beschäftigen sich mit Dateien sowie mit der Programmierung von Grafik und Betriebssystemfunktionen. Auf diese Weise wird dem Neuling vom ersten Einzeller bis zu komplexen Programmen die Arbeit mit diesem komfortablen BASIC-Dialekt nahegebracht. ▶ Viele Beispielprogramme, die teilweise, wie etwa ein Fakturierungsprogramm, explizit entwickelt werden, runden das Buch ab. Damit die einzelnen Programme verständlicher werden, sind sie, wo nötig, durch Ablaufdiagramme ergänzt. Um nicht alle Listings abtippen zu müssen, liegt dem Buch eine Diskette mit allen Programmen bei. ▶ Aber auch, wenn man BASIC bereits beherrscht, ist das Buch durch seine zahlreichen Anhänge, die unter anderem eine ASCII-Tabelle, eine Übersicht der Füllmuster und BITBLT-Modi sowie ein Verzeichnis der VT52-Codes und einen Index enthalten, als Nachschlagewerk wertvoll. Zu diesem Zweck wurde auch eine vollständige alphabetisch sortierte Kurzübersicht der Befehle integriert.

INHALT:
▶ Erklärung der Schleifen- und Programmstrukturen – *Primzahlenberechnung – Zahlenraten* ▶ Variablentypen und Arrays – *Sieb des Eratosthenes – Adreßeingabe* ▶ Numerische und Stringfunktionen – *Alle trigonometrische Funktionen wie Sinus, Cosinus etc. – Die Zufallsfunktion* ▶ Unterprogramme und Prozeduren – *Rekursive Prozeduraufrufe – Suche des Ausgangs aus einem Labyrinth* ▶ Grafikprogrammierung – *Grafische Grundelemente wie Linie, Kreis, Rechteck – Blockoperationen – Flimmerfreie Animation durch Einsatz mehrerer Bildschirme – Erzeugen eines Balkendiagramms* ▶ Multitasking in Omikron-BASIC – *Druckerspooler* ▶ Programmierung von Abstrakten Datentypen – *Die Datenstruktur „Schlange" – Verkettete Listen* ▶ Dateiverwaltung – *Programmierung von sequentiellen Dateien – Arbeiten mit Random-Access-Dateien* ▶ Programmprojekt Fakturierung – *Ausführliche Entwicklung eines Programms auf ca. 30 Seiten* ▶ Betriebssystemprogrammierung – *Einsatz von Alertboxen und der Fileselectbox – Benutzen von Pulldown-Menüs, Fenstern und Dialogboxen – Programmierung einer Druckeranpassung in BASIC* ▶ Programmentwicklung und Debugging – *Fehlersuche und -beseitigung* ▶ Sammlung von Beispielprogrammen – *Ausgabe eines Diskettenverzeichnisses – Backup-Programm für die Harddisk – Turtle-Grafik in Omikron-BASIC – Adreßverwaltung* ▶ Alphabetische Befehlsübersicht



Hardcover
über 400 Seiten einschl. Programm-Diskette

DM 59, – Best.-Nr. B-413

Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle: ______St. Das große OMIKRON-BASIC-BUCH (incl. Programm-Diskette) á DM 59,-
______St. Programmieren in OMIKRON-BASIC á DM 49,-
______St. PROGRAMMDISKETTE zum Buch á DM 39,-
______St. Kurz & Klar Nachschlagewerk OMIKRON-BASIC á DM 29,-

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

und Faktorenanalyse sowie - als einziges Programm - die Diskriminanzanalyse.

Bei der alphanumerischen und graphischen Ausgabe (siehe Abb. 5 bis 7) sind die wichtigsten Funktionen enthalten, besonders gelungen ist dabei u.a. die graphische Darstellung der Faktorenanalyse. Lediglich dreidimensionale Graphiken fehlen bisher, die Autoren kündigen jedoch für die nahe Zukunft hierfür Abhilfe durch ein weiteres Programm ("proGraph") an.

OMIKRON-BASIC STATISTIK LIBRARY

Das letzte hier besprochene Statistikprogrammpaket benötigt, wie bereits erwähnt, den OMIKRON.BASIC-Interpreter. Als Bibliothek von Routinen, die in eigene BASIC-Programme eingebunden werden können, ist es mit den anderen besprochenen Programmen nicht direkt vergleichbar (und deshalb nicht in der Übersichtstabelle aufgeführt).

Die Statistik-Library erfordert eine gewisse Programmiererfahrung, bietet dafür jedoch sehr preiswert etliche nützliche und erweiterbare Prozeduren an, u.zw.:

- Zufallsgeneratoren
- Datentransformation
- Grundstatistiken
- Verteilungs- und Prüffunktionen
- Konfidenzintervalle
- Anpassungstests
- Vergleich zweier Stichproben
- lineare und multiple Regression, Korrelation
- Kontingenztabellen
- Chi-Quadrat (χ^2)-Test und Kolmogorov-Smirnov-Test
- zweifaktorielle Varianzanalyse
- Zeitreihenanalyse.

Neben zwei einfachen Routinen für die Dateneingabe stellt diese Bibliothek auch einige Routinen für die Grafikausgabe zur Verfügung (Punkte-, Linien-, Block- sowie Kreis-Diagramme). Alles in allem für den in BASIC-Programmierung erfahrenen Statistiker ein lohnenswertes und sehr preiswertes Programmpaket.

ST-STATISTIK 2.6

Autor / Vertrieb: Michael Prall / SciLab GmbH, Isestr. 57, 2000 Hamburg 13

Hardware-Voraussetzung: ATARI ST ab 1 Megabyte RAM, mind. 1 Diskettenlaufwerk mit 720 kByte, S/W-Monitor

Programmiert in: GFA-BASIC

Kopierschutz: Paßwort

Handbuch: Ringordner DIN A5, ca. 88 Seiten (knapp, aber aufgrund der Benutzerfreundlichkeit des Gesamtprogramms für Benutzer mit Statistikkenntnissen ausreichend; Angabe von Beispielen, Literaturangaben)

Preis: DM 349.-

VARIANA

Autor / Vertrieb: Christoph Zielinski / G/P-Elektronik AG, Schoenleinstr. 12, 1000 Berlin 61

Hardware-Voraussetzung: ATARI ST ab 512 kByte, mind. 1 einseitiges Diskettenlaufwerk, S/W-Monitor

Programmiert in: C

Kopierschutz: Hardware-Laufmodul im ROM-Port

Handbuch: Ringordner DIN A5, ca. 150 Seiten (ausführlich, Angabe von Formeln, Beispielen, Literaturangaben)

Preis: DM 539.-

WISTAT 2.0, WISTAT GRAPH / REGRESSION ST 3.0

Autor / Vertrieb: Ule Franzen bzw. H.W. Fritsch / tel soft, Universitätsstr 40, 3550 Marburg

Programmiert in: GFA-BASIC (WiSTat)
ST-PASCAL +, ASSEMBLER (Regression ST, WiSTat Graph)

Hardware-Voraussetzung: ATARI ST oder ATARI Mega ST, keine Voraussetzungen bzgl. des Diskettenlaufwerks, S/W-Monitor

Kopierschutz: keiner, jedoch Seriennummer und Anwendername mehrfach im Programm verschlüsselt eingetragen

Handbuch (WiSTat): Ringbuch DIN A5, ca. 86 Seiten (ebenfalls knapp, aber aufgrund der Benutzerfreundlichkeit des Gesamtprogramms für Benutzer mit Statistikkenntnissen ausreichend; Angabe von Beispielen, Formeln, Literaturangaben)

Preise: WiSTat DM 199.-
WiSTat Graph DM 99.-
Regression ST DM 198.-

OMIKRON-BASIC STATISTIK LIBRARY (STATLIB)

Autor / Vertrieb: Jörn Wilms / OMIKRON.Software, Erlachstr. 15, 7534 Birkenfeld 2

Hardware-Voraussetzung: ATARI ST oder ATARI Mega ST

Software-Voraussetzung: OMIKRON.BASIC

Kopierschutz: keiner Handbuch (WiSTat): Heft DIN A5, 11 Seiten, (Kurzbeschreibung, Literaturangaben, Statistikkenntnisse und Programmiererfahrung erforderlich)

Preis: DM 79.-

Abschließende Bemerkungen

Es wäre vermessen zu behaupten, die obige Übersicht habe alle Statistikpakete in all ihren Möglichkeiten voll berücksichtigt; dies ist allein aus Platzgründen nicht möglich. Weiterhin werden die gleichen Funktionen oft nach ganz unterschiedlichen Prinzipien realisiert, und die Leichtigkeit der Handhabung ist oft nicht nur vom Programm, sondern auch von dem Ziel des jeweiligen Anwenders abhängig. Deshalb werden hier auch keine "Noten" vergeben, sondern die Empfehlung ausgesprochen, daß sich potentielle Interessenten solcher Programme an einen Händler wenden und die Programme für ihre Zwecke erst einmal selbst testen sollten. Hierfür kann dieser Artikel allerdings - so hoffe ich - eine wertvolle Orientierung und Einstiegshilfe sein.

Diese Empfehlung ist auch deshalb nützlich, weil einige dieser Programme (besonders ST STATISTIK und WISTAT) in rascher Folge Updates mit zusätzlichen Prozeduren und Verbesserungen auf den Markt brachten und dies vermutlich auch in der nahen Zukunft tun werden. Einige (kleinere) Kritikpunkte und Empfehlungen seien jedoch noch erwähnt:

Auch die neuesten Versionen der vorgestellten Programme enthalten noch kleine Fehler, die manchmal (selten !) zum Programmabsturz führen. Dies gilt v.a. für die beiden sehr komplexen Programmpakete ST STATISTIK und WISTAT. Ich habe bei früheren Vorversionen die Autoren selbst auf etliche (z.T. "verborgene") Fehler aufmerksam gemacht, wobei die Autoren dieser beiden Programmpakete erfreulich schnell und sorgfältig auf die Hinweise reagierten und umgehend Korrekturen durchführten. Nach ausführlichem Testen dieser Programme kann ich trotz der noch bestehenden kleineren Fehler bestätigen, daß sich mit den hier vorgestellten Programmen sehr effektiv und sehr komfortabel arbeiten läßt.

Inwieweit die Handbücher dem Anwender ausreichen, hängt aufgrund der allgemeinen Bedienungsfreundlichkeit der Programme vor allem vom Kenntnisstand des Anwenders in Statistik ab. Gerade für den Anfang dürften bei Neulingen auf diesem Gebiet folgende Punkte wichtig sein (sind in den Handbüchern teilweise erfüllt):

(a) Angabe der Formel oder des genauen Algorithmus' für die Prozedur, (b) Angabe der für die Prozedur relevanten Literatur (insbesonders bei Übernahme oder Adaptation von Programmen von anderen Autoren) sowie (c) ein oder mehrere Berechnungsbeispiele.

Fazit

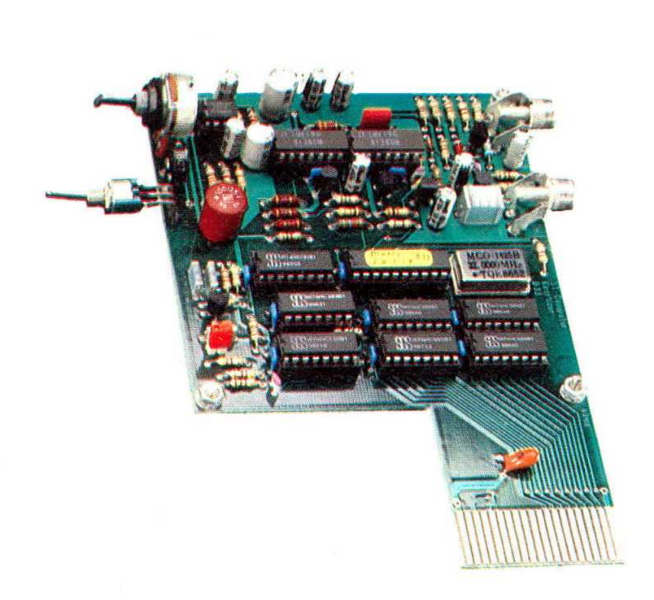
Für die wissenschaftliche Statistik stehen nunmehr auch für den ATARI ST sehr leistungsfähige, grafikfähige, leicht bedienbare und nicht zuletzt sehr preiswerte Pakete zur Verfügung, die dem Anwender mit statistischen Vorkenntnissen eine überaus große Hilfe bei der Auswertung wissenschaftlicher Daten sein können.

P.S.: Da immer mehr Software-Pakete für den Einsatz des ATARI ST im wissenschaftlichen Bereich entwickelt werden, soll in "ST COMPUTER" auch künftighin über diese Entwicklungen, Updates und Erfahrungen berichtet werden. Um dies zu erleichtern, werden diejenigen, die den ATARI ST wissenschaftlich einsetzen und neue Programme entwickelt haben, gebeten, uns ihre Produkte zur Information zuzusenden.

Arthur Günthner



Hardware, die sich bewährt hat !!!







Easytizer - der Videodigitizer ohne Geheimnisse

Mit dem Easytizer können Sie beliebige Videosignale von einer Schwarzweiß- oder Farb-Kamera, Videorecorder oder direkt vom Fernsehgerät (mit Composite Video Ausgang) digitalisieren und somit auf dem Bildschirm Ihres ATARI ST sichtbar machen.
Der Easytizer wird am Modul-Port des ST angeschlossen. Die Auflösung beträgt 800x600 Bildpunkte, so daß in mittlerer Auflösung 640x200 Bildpunkte in vier Graustufen dargestellt werden können. In dieser Betriebsart werden 12,5 Bilder in der Sekunde wiedergegeben. Im hochaufgelösten Modus werden 640x400 Bildpunkte in Schwarzweiß wiedergegeben.

Besondere Merkmale des Easytizers:

- Software vollständig in Assembler
- Abspeichern der Bilder im DEGAS-Format
- Von STAD und Sympatic Paint ansteuerbar
- Ein beliebiger Bildausschnitt kann in ein mit der Maus wählbares Format vergrößert und verkleinert werden
- Spiegeln eines Bildes in horizontaler und vertikaler Ebene
- Animation mit beliebig vielen Bildern möglich, nur durch die Kapazität des Rechners begrenzt (beim Mega ST4 über 100 Bilder)
- Eingebauter Druckertreiber für die mittlere Auflösung für NEC P6/P7 und EPSON oder Kompatible
- Wahlweise automatische oder manuelle Helligkeitseinstellung
- Schnappschuß

Lieferumfang:

1) Fertiggerät
   komplett aufgebaut und geprüft, inclusive Diskette mit der Easytizer Software und Bedienungsanleitung
2) Teilsatz
   Für Bastler liefern wir einen Teilsatz bestehend aus:
   - Doppelseitiger, elektronisch geprüfter Platine mit Lötstoplack und Bestückungskungsaufdruck sowie vergoldeten Anschlußkontakten
   - fertig programmiertes GAL 16V8
   - Quarzoszillatormodul 32 MHz
   - Diskette undBedienungsanleitung.

Klein, kompakt und leistungsstark der *Junior Prommer*

Der Junior Prommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 kByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMs, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren.
Zum Betrieb benötigt der Junior Prommer nur +5 Volt, die am Joystick-Port Ihres ATARI ST abgenommen werden, alle anderen Spannungen erzeugt die Elektronik des Junior Prommers. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen.
Selbstverständlich läßt sich ein 16-Bit Word in ein High- und Low-Byte zerlegen. Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex/ASCII-Monitor läßt sich der Inhalt eines EPROMs blitzschnell durchsuchen oder auch ändern.

Alles dabei!

Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertiggerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlußfertig geliefert. Auf der Diskette mit der Treibersoftware befinden sich noch RAM-Disk und ein Programm zum Erstellen von EPROM-Karten, ferner wird der Source-Code für Lese- bzw. Programmierroutinen mitgeliefert und last but not least ist im Bedienungshandbuch der Schaltplan abgedruckt.

Typ

INT. ID.			
EPROM'S			
2716	25V		
2732	25V	2732A	21V
2764	21V	2764A	12V
27128	21V	27128A	12V
27256	12V	27256	21V
27512	12V	27513	12V
27011	12V		
ROM'S			
4732		4764	
47128		47256	
EEPROM'S			
X2804A		X2816A	
X2864A		X28256A	

Übersicht der mit dem Junior Prommer programmierbaren Speichertypen

Textverarbeitung mit TEMPUS

Die Position von TEMPUS als schnellster und praktisch in jeder Hinsicht bester Editor für den ATARI ST ist weitgehend unumstritten. Ganz anders sieht es hingegen aus, wenn TEMPUS als Textverarbeitungsprogramm "mißbraucht" werden soll - seine wenigen, eher primitiven Kommandos zur Druckersteuerung erlauben scheinbar wenig mehr als den vorhandenen Text so auszudrucken, wie er am Bildschirm dargestellt wird.

Um einzelne Wörter innerhalb eines Textes hervorzuheben, kann nicht nur *Kursivdruck*, **Fettdruck**, Unterstreichen oder ***eine Kombination*** davon eingesetzt werden - auch die Änderung der Zeichen-breite oder des Zeichenabstands ist möglich. Der perfekte Blocksatz in Proportionalschrift und mit weichen Trennungen (ohne zentimetergroße leere Zwischenräume) wird dabei keineswegs gestört.

Erganzende Miteilungen werden in Kleinstschrift, seitlich etwas eingerückt und mit verringertem Zeilenabstand gedruckt. Dies ist beispielsweise bei der Formulierung der wichtigen Passagen eines Versicherungsvertrages (die der potentielle Kunde moglichst gar nicht oder erst nach Vertragsabschluß lesen soll) von größter Bedeutung.

Abb. 1: Beispielausdruck

Mit etwas Phantasie und einem "intelligenten" Drucker läßt sich das beheben: eleganter Blocksatz mit beliebiger Spaltenbreite, verschiedene Zeichengrößen innerhalb einer Zeile, variabler Zeilenabstand,.. sind mit modernen Druckern durchaus realisierbar! Teilweise sind die Gestaltungsmöglichkeiten mit TEMPUS größer als mit weitverbreiteten Textverarbeitungsprogrammen - der Beispielausdruck (Abb. 1) beweist es.

Die hier am Beispiel des NEC P6 beschriebene Druckerinstallation geht davon aus, daß der Text "in einer Wurst" ausgedruckt wird und sämtliche notwendigen Formatierungen vom Drucker vorgenommen werden. Steuercodes an den Drucker werden als Sonderzeichen im Text integriert. Um dabei im Text nicht die Übersicht zu verlieren, empfiehlt es sich, einen eigenen Zeichensatz zu verwenden, der die Steuerzeichen als möglichst eindeutige Symbole enthält.

Blocksatz

Der erste Schritt zu einem ordentlich "gestylten" Text besteht darin, das Problem des Randausgleichs zu lösen. Der von TEMPUS verwendete Randausgleich ist hierfür ungeeignet, weil in ihm unterschiedliche Zeichenbreiten nicht berücksichtigt werden können - das ist aber vor allem bei Proportionalschrift wichtig. Zudem würden im Text integrierte Steuerzeichen (beispielsweise zur Umschaltung auf Kursivschrift) den Blocksatz zerstören.

Es wird daher der Drucker in den Justierungsmodus "Blocksatz" versetzt. Der Drucker versucht nun, den ankommenden Text möglichst optimal in der Zeile aufzuteilen. Dazu werden Leerräume zwischen den Wörtern variiert. Sobald ein Wort nicht mehr vollständig in einer Zeile Platz hat, wird die nächste Zeile begonnen. Es kann daher der gesamte Absatz ohne weitere Steuercodes am Zeilenende gedruckt werden - der Drucker sorgt selbständig für die Formatierung!

Dieses Verfahren ist aber noch nicht optimal: Es wird nämlich auch bei der jeweils letzten Zeile eines Absatzes versucht, diese im Blocksatz zu drucken, was unerwünscht ist - diese Zeile soll linksbündig gedruckt werden. Auch dieses Problem ist lösbar: beim Ausdruck des Absatzendezeichen (CR = Carriage Return) wird der Drucker nicht einfach zum Ausdruck der Zeile und zum Zeilenvorschub veranlaßt. Stattdessen wird er vorher in den Justierungsmodus "linksbündig" versetzt: Die zu diesem Zeitpunkt noch nicht gedruckte Zeile befindet sich im druckerinternen Buffer und wird nun neu formatiert. Erst jetzt folgt der Code für den Zeilenvorschub und Wagenrücklauf - die Zeile wird (linksbündig) gedruckt. Anschließend folgen einige Codes, die den Drucker (für den nächsten Absatz) wieder in den Blocksatzmodus versetzen. Auf diese Weise ist auch sichergestellt, daß einzelne Zeilen (z.B. Überschriften) immer linksbündig gedruckt werden.

Soll erreicht werden, daß die jeweils erste Zeile jedes Absatzes um einige Zeichen eingerückt wird, können in der Installationsdatei an die Codes für das CR-Zei-

```
Beispielausdruck

Um einzelne Wörter innerhalb eines Textes hervorzuheben, kann
nicht nur Kursivdruck, Fettdruck, Unterstreichen oder
eine Kombination davon eingesetzt werden - auch die Änderung
der Zeichenbreite oder des Zeichenabstands ist möglich.
Der perfekte Blocksatz in Proportionalschrift und mit weichen
Trennungen (ohne zentimetergroße leere Zwischenräume) wird
dabei keineswegs gestört.

Ergänzende Miteilungen werden in Kleinstschrift, seitlich
etwas eingerückt und mit verringertem Zeilenabstand gedruckt.
Dies ist beispielsweise bei der Formulierung der wichtigen
Passagen eines Versicherungsvertrages (die der potentielle Kunde
möglichst gar nicht oder erst nach Vertragsabschluß lesen soll)
von größter Bedeutung.
```

Abb. 2: Der zum Beispielausdruck gehörende Originaltext mit Steuerzeichen

chen einige Leerzeichen (in Hochkommata) angefügt werden. Es besteht auch die Möglichkeit, einzelne Zeilen zentriert oder rechtsbündig zu drucken (Überschriften, Datum) - dazu finden sich weiter unten genauere Angaben.

Worttrennungen

So richtig perfekt wirkt ein Text in Blocksatz erst dann, wenn die Leerräume zwischen den Wörtern nicht zu groß werden. Dazu kann es aber kommen, wenn ein langes Wort in der Zeile gerade nicht mehr Platz hat. Bei guten Textverarbeitungsprogrammen wird dieser Fall durch eine automatische Silbentrennung vermieden. Dies ist in TEMPUS nicht möglich - es gibt aber eine Alternative: sogenannte weiche Trennungen.

Wird ein längeres Wort geschrieben, werden zur Sicherheit an den richtigen Stellen Trennungszeichen miteingegeben, also beispielsweise "Text~verarbeitungs~programm" oder "Programm~entwicklungs~umgebung". Beim Ausdruck sollen diese Trennungszeichen natürlich nicht gedruckt werden - außer das Wort muß wirklich getrennt werden. Für ein gleichmäßiges Erscheinungsbild des Textes ist es ausreichend, diese weichen Trennungen nur bei wirklich langen Worten vorzusehen - daher ist der zusätzliche Tippaufwand minimal. Damit der Bindestrich aber auch weiterhin als solcher verwendet werden kann, sollte als Trennungszeichen ein anderes Zeichen - beispielsweise die Schlangenlinie ~ benutzt werden.

Wie kann nun der richtige Ausdruck dieser Trennungszeichen erreicht werden? Die Idee dazu ist relativ einfach und basiert darauf, daß eine Zeile erst vollständig im internen Puffer des Druckers gespeichert und verarbeitet werden muß, bevor sie ausgedruckt werden kann (sonst wäre beispielsweise Blocksatz unmöglich). Beim Ausdruck des Trennungszeichens wird getestet, ob in der Zeile noch genug Platz für den Rest des Wortes bleibt oder ob das Wort hier getrennt werden soll. Je nachdem, wie dieser Test ausfällt, wird das Trennungszeichen gedruckt oder nicht gedruckt.

Zuerst ein relativ gut verständlicher (aber leider nicht realisierbarer) Vorschlag, wie dieser Test bewerkstelligt werden kann: Statt des Trennungszeichens wird "- XXXXX" und anschließend 7mal der Code 127 ausgegeben. Im Normalfall werden die sieben Zeichen "- XXXXX" durch die 127-Codes (Kommando "Zeichen löschen") wieder aus dem Puffer gelöscht und nicht gedruckt.

Nur wenn diese Zeichenkombination am Zeilenende auftritt und "XXXXX" dort nicht mehr Platz hat, kommt es beim Leerzeichen hinter dem Trennungszeichen zu einem Zeilenumbruch: Das Trennungszeichen wird als letztes Zeichen der Zeile gedruckt. Die fünf "X" kommen in den Zeilenbuffer für die nächste Zeile, werden dort aber durch die 127-Codes gelöscht (und somit nicht ausgedruckt). Daß dabei zwei 127-Codes zuviel auftreten, stört nicht.

Das Verfahren produziert aber dennoch (aus unbekanntem Grund) in etwa 10% der Fälle Fehler - das fragliche Wort wird zwar getrennt, aber der Trennungsstrich wird nicht wie vorgesehen gedruckt, dafür werden in der nächsten Zeile einige Zeichen übereinander gedruckt.

Nach langem Probieren wurde schließlich eine ähnliche Methode gefunden, die fehlerfrei funktioniert. Es wird wiederum das Trennungszeichen ausgegeben, nun folgen aber Codes, die den Druckkopf in mehreren Schritten um insgesamt $^{99}/_{180}$ Zoll nach rechts bewegen (Details hierzu etwas weiter unten). Es wird nun ein "X" ausgegeben. Dieses hat wie beim obigen Verfahren eine reine Testfunktion. Es können jetzt zwei Fälle auftreten:

(1) Wenn sich der Druckkopf am Zeilenende befindet, erfolgt ein Zeilenumbruch: Die 99/180 Zoll wirken wie ein Leerzeichen - als letztes Zeichen der Zeile wird also der Bindestrich ausgegeben (rechtsbündig). Das "X" kommt in den Puffer für die nächste Zeile und wird durch den ersten 127-Code gelöscht. Nun wird versucht, den Wagen um $^{99}/_{180}$ Zoll (nach links) zurückzubewegen und mit dem 127-Code ein weiters Zeichen zu löschen - beide Kommandos können nicht ausgeführt werden, weil der Druckkopf bereits am Zeilenanfang (der nächsten Zeile) steht. Somit wird der Druck am Beginn dieser Zeile fortgesetzt - eine korrekte Trennung wurde durchgeführt.

(2) Wenn das "X" in der oberen Zeile noch Platz hat, kommt es zu keinem Zeilenumbruch - stattdessen wird zuerst das "X" gelöscht (erster 127-Code), dann der Wagen um $^{99}/_{180}$ Zoll zurückbewegt und schließlich auch das Trennungszeichen wieder gelöscht (zweiter 127-Code). Somit ist im Ausdruck vom Trennungszeichen nichts zu sehen. Damit ist exakt der gewünschte Effekt eingetreten: Wenn es notwendig ist, wird das Wort getrennt und das Trennungszeichen korrekt ausgedruckt, andernfalls wird das Wort ohne Unterbrechung und ohne Bindestrich gedruckt.

Ein ähnliches Problem tritt übrigens auch beim normalen Bindestrich auf, wenn dieser in Wortzusammensetzungen verwendet wird. Der Drucker führt beim Bindestrich nämlich keine Worttrennung durch, falls diese notwendig wäre. Der Lösungsweg ist derselbe wie oben beschrieben, nur der letzte 127-Code fällt weg (weil der Bindestrich in jedem Fall gedruckt werden soll).

Die Codes des Wagentransportkommandos lauten beim NEC P6: *27 92 N1 N2*. Der Wagen wird um *N1+N2*256* $^{1}/_{180}$ Zoll weitertransportiert. Für negative Werte muß das Zweierkomplement verwendet werden - für minus 99 also: 157 255 bzw. hexadezimal $A6 $FF.

Nun ist nur noch unklar, warum der Druckkopf in nicht weniger als sieben Schritten (50+25+12+6+3+2+1) um $^{99}/_{180}$ Zoll nach vorn bewegt wird. Der Grund ist einfach: Würde versucht, den Druckkopf um $^{99}/_{180}$ Zoll weiterzubewegen, obwohl nicht mehr ausreichend Platz ist (was am Zeilenende oft der Fall ist), würde das Kommando einfach ignoriert. Durch die vorliegende Kombination wird aber erreicht, daß der Druckkopf auch in diesem Fall garantiert bis ans Zeilenende weiterbewegt wird.

Noch ein Hinweis: In der Druckerinstallationsdatei für TEMPUS dürfen die Codes für die Definition eines Zeichens ohne weiteres über mehrere Zeilen reichen (zumindest in Version 2.05).

Seitengestaltung

Dies ist eindeutig der schwächste Punkt der Installation - da die Seitenlänge (bei Verwendung verschiedener Zeichensätze) nicht vorhersehbar ist, können Kopf- und Fußzeilen nicht verwendet werden.

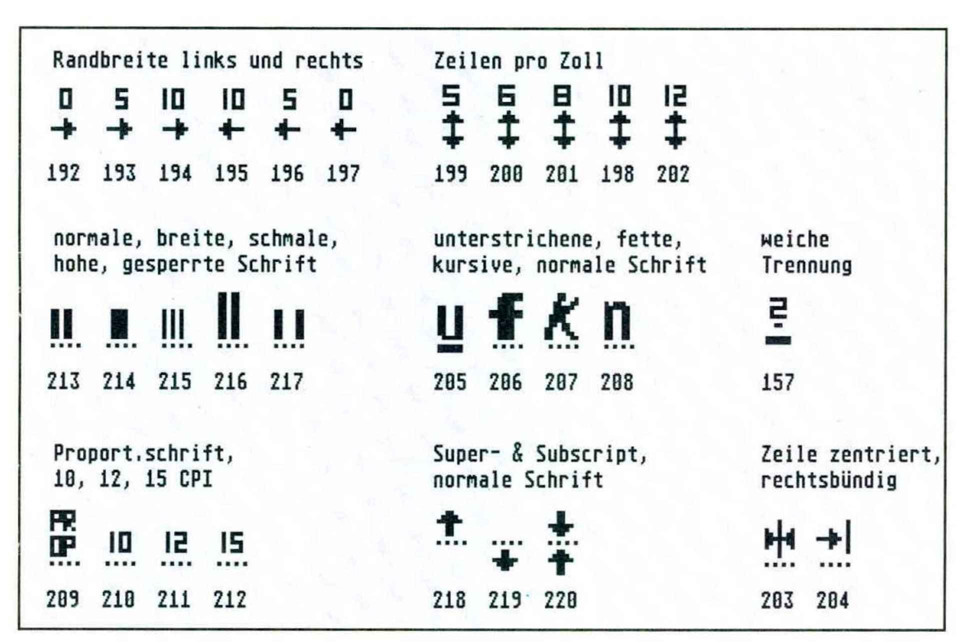


Abb. 3: Symbolik der Steuerzeichen

Achtung: Der hier vorgestellte Lösungsweg ist (naturgemäß) nicht perfekt. Wenn das Wortende länger als $^{99}/_{180}$ Zoll (bei Proportionalschrift etwa 7 bis 8 Kleinbuchstaben) ist, kann folgender Fall eintreten: Es sind am Zeilenende noch $^{99}/_{180}$ Zoll Platz (die Trennung wird daher nicht durchgeführt), dieser Raum reicht aber nicht für das Wortende. Daher wird das gesamte Wort in die nächste Zeile verschoben, die unerwünscht großen Wortzwischenräume (in der oberen Zeile) treten erneut auf. Der Ausweg besteht darin, statt $^{99}/_{180}$ Zoll einen größeren Freiraum zu verwenden (der neue Wert muß natürlich analog in Einzelwerte zerlegt werden; am einfachsten wäre dies für 127 möglich: 64+32+16+8+4+2+1).

Allerdings kann nun der Fall eintreten, daß ein Wort noch vollständig Platz hätte, aber dennoch getrennt wird (weil der Freiraum keinen Platz hat). Ich empfinde 99/180 Zoll (bei Proportionalschrift) als einen guten Kompromiß, eigenes Experimentieren ist aber immer lohnend.

Eine neue Seite beginnt einfach dann, wenn auf der letzten Seite kein Platz mehr ist. Es kann lediglich durch FF-Zeichen (Form-Feed) der Beginn einer neuen Seite (z.B. bei einem Kapitelanfang) vorgegeben werden.

Ganz so schlimm, wie das hier klingt, ist es aber auch wieder nicht. Mit Hilfe des sogenannten Perforationssprungs kann erreicht werden, daß die obersten und untersten Zeilen der Seite nicht bedruckt werden. Ein Perforationssprung von 8 bedeutet, daß die letzen 8 Zeilen einer druckerinternen Seite nicht bedruckt werden, sondern das Papier entsprechend weitertransportiert wird. Wird nun der Drucker so eingeschaltet, daß der Seitenbeginn des Papiers bereits einige Zeilen über dem Druckkopf steht, werden sowohl am Seitenanfang als auch am -ende einige Zeilen freigehalten.

Noch etwas kann den Ausdruck längerer Texte erheblich stören: Überschriften oder die erste Zeile eines neuen Absatzes werden ausgerechnet am Seitenende gedruckt. Dieser Umstand ist behebbar: Beim Druck des CR-Zeichens (das ja nur am Absatzende vorkommt) wird der Perforationssprung zuerst auf 11 Zeilen erhöht, dann folgt der Code für den Zeilenvorschub und Wagenrücklauf, und schließlich wird der Perforationssprung wieder auf 8 Zeilen reduziert. Dadurch wird am Absatzende auch dann ein Papiertransport zur nächsten Seite ausgeführt, wenn eigentlich noch drei bedruckbare Zeilen frei wären. (Die drei Zeilen werden aber sehr wohl verwendet, wenn ein neuer Absatz schon eine Zeile weiter oben beginnt.)

Sonstige Steuercodes

Alle weiteren Codes, etwa um den Drucker auf Kursivschrift umzustellen, um hochgestellt zu drucken, den Zeilenabstand zu ändern, die Zeichenbreite zu ändern,... können einfach aus dem Druckerhandbuch entnommen werden - es sind keine weiteren Tricks mehr notwendig. Zur Anwendung dieser Codes sind aber noch einige Hinweise angebracht:

Die Kommandos "hoch-" und "tiefstellen" sind nicht nur für Indizes und Hochzahlen einsetzbar. Es kann auch ein ganzer Absatz in dieser Kleinschrift gedruckt werden. Am besten wirkt dies bei Proportionalschrift, weil dann auch der Abstand der Buchstaben (entsprechend ihrer Kleinheit) reduziert wird. Außerdem empfiehlt es sich, die Zahl der Zeilen pro Zoll auf 10 oder 12 zu erhöhen. Damit kann man inhaltlich nicht so wichtige Ergänzungen, weitergehende Erläuterungen,.. sehr gut vom restlichen Text abgrenzen (sozusagen als Ersatz für die nicht vorhandene Möglichkeit von Fußnoten; siehe auch Abb. 1).

Das Kommando "Fettschrift" ist beim NEC P6 nicht besonders wirkungsvoll. Daher wurde es bei der vorliegenden Installation mit "schattiertem Druck" kombiniert - der so gekennzeichnete Text **sticht wirklich ins Auge.** Relativ unbekannt ist das Kommando, um den Abstand der Zeichen eines Wortes einzustellen - damit kann ein g e s p e r r t e r Druck erreicht werden (zwischen den Buchstaben wird ein beliebiger Zwischenraum - z.B. ein viertel Leerzeichen - freigelassen, um das Wort so hervorzuheben).

Zu den Kommandos "Schmalschrift" und "d o p p e l t e S c h r i f t b r e i t e" ist zu erwähnen, daß sie einander keines-

wegs kompensieren. Werden beide Kommandos zugleich eingesetzt, ergibt sich eine "halbbreite" Schrift, die etwas größer als die normale ist. Für Überschriften ist eine Kombination von "Fettschrift", "doppelte Schriftbreite" und "doppelte Schrifthöhe" sehr wirkungsvoll.

Der Blocksatz wird übrigens nicht gestört, wenn innerhalb einer Zeile unterschiedliche Zeichenbreiten verwendet werden - der NEC P6 ist hier (im Gegensatz zu vielen Textverarbeitungsprogrammen) sehr flexibel. Wenn das Kommando "doppelte Schrifthöhe" für ein Wort innerhalb eines Absatzes verwendet wird, wird automatisch eine zusätzliche Leerzeile gedruckt, um Platz für die großen Buchstaben zu schaffen.

Proportionalschrift ist zwar in der Regel optisch ansprechender als Schrift mit gleichbleibender Buchstabenbreite. Für Tabellen muß aber dennoch eine nicht proportionale Schrift (10, 12 oder 15 CPI = Character per Inch) verwendet werden, weil sonst die variable Buchstabenbreite den Tabellenaufbau stört.

Mit den Kommandos "linker Rand" und "rechter Rand" kann bestimmt werden, wie breit der Papierrand an beiden Seiten des Ausdrucks sein soll. Es besteht damit auch die Möglichkeit, einzelne Absätze stärker als den übrigen Text einzurücken (siehe Beispielausdruck). Dabei ist zu beachten, daß diese Kommandos je nach Zeichenbreite unterschiedliche Resultate zeigen. Für die Installation wurden 10 CPI angenommen - daher sollten die Kommandos nur bei dieser Zeichenbreite oder bei Proportionalschrift (dort wird ebenfalls mit 10 CPI gerechnet) verwendet werden. Gegebenfalls muß die Schriftbreite also für die Ausführung dieser Kommandos kurz geändert werden. Die Kommandos dürfen nur am Zeilenanfang verwendet werden!

Hierzu noch ein Hinweis: Beim Beispielausdruck (Abb. 1) wurde der rechte Rand nicht auf 70, sondern auf 47 Zeichen gesetzt (daher wurde der kurze Text in mehreren kurzen und nicht in wenigen langen Zeilen ausgedruckt). Die Kommandos "rechtsbündig" und "zentriert" gelten nur für eine Zeile. Die Steuerzeichen müssen vor und (!) hinter dem entsprechenden Text verwendet werden; es darf kein Absatzendezeichen (CR) dazwischen liegen, weil bei der Ausführung der Codes dieses Zeichens kurzzeitig auf linksbündigen Druck umgestellt wird (siehe oben). Die richtige Anwendung ist

```
 1:  * TEMPUS-Druckerinstallation für den NEC P6
 2:  * Michael Kofler, Mai 89
 3:
 4:  VORTEXT:= 27,64, 27,78,8, 27,97,3
 5:  *   Drucker initialisieren, Perforation 8 Zeilen, Blocksatz
 6:
 7:  NACHTEXT:= 27,64   *Drucker neu initialisieren
 8:  VORSEITE:=         *keine Codes bei Tempus-internem Seitenumbruch
 9:  NACHSEITE:=
10:  VORZEILE:=         *keine Codes bei Tempus-internen Zeilenumbruch
11:  ENDZEILE:=
12:
13:  "P":=225           *scharfes S durch griechisches Beta darstellen
14:
15:  13:=27,97,0, 27,78,11,  10,  27,78,8, 27,97,3
16:  *   CR-Zeichen (am Absatzende):
17:  *    letzte Zeile des Absatz linksbündig drucken,
18:  *    Perforation kurzzeitig 11 Zeilen, Zeilensprung,
19:  *    Perforation wieder 8 Zeilen, wieder Blocksatz
20:
21:
22:  157:="-", 27,92,50,0, 27,92,25,0, 27,92,12,0, 27,92,6,0, 27,92,3,0,
23:       27,92,2,0, 27,92,1,0,  "X",127,  27,92,157,255, 127
24:  *    weiche Trennung:
25:  *    Bindestrich, Druckkopf 50+25+12+6+3+2+1 * 1/180 nach vor, "X",
26:  *    mit 127 das "X" wieder löschen, Druckkopf um 99/180 zurück,
27:  *    mit 127 den Bindestrich löschen
28:  *    genaue Erklärung der Codes im Text
29:
30:  45:="-", 27,92,50,0, 27,92,25,0, 27,92,12,0, 27,92,6,0, 27,92,3,0,
31:       27,92,2,0, 27,92,1,0,  "X",127,  27,92,157,255
32:  *   Bindestrich: wie weiche Trennung, aber ohne den 127-Code am Ende -
33:  *   der Bindestrich wird also nicht gelöscht
34:
35:  192:=27,108,0                          *Linker Rand nicht eingerückt
36:  193:=27,108,5                          *Linker  Rand 5 Zeichen eingerückt
37:  194:=27,108,10                         *Linker  Rand 10 Zeichen eingerückt
38:
39:  195:=27,81,70                          *Rechter Rand 10 Zeichen eingerückt
40:  196:=27,81,75                          *Rechter Rand 5 Zeichen eingerückt
41:  197:=27,81,80                          *Rechter Rand nicht eingerückt
42:
43:  199:=27,65,12                          *5 Zeilen pro Zoll
44:  200:=27,50                             *6 Zeilen pro Zoll
45:  201:=27,48                             *8 Zeilen pro Zoll
46:  198:=27,65,6                           *10 Zeilen pro Zoll
47:  202:=27,65,5                           *12 Zeilen pro Zoll
48:
49:  209:=27,112,1                          *Proportionalschrift
50:  210:=27,112,0, 27,80                   *Prop. aus, 10 Zeichen pro Zoll
51:  211:=27,112,0, 27,77                   *Prop. aus, 12 Zeichen pro Zoll
52:  212:=27,112,0, 27,103                  *Prop. aus, 15 Zeichen pro Zoll
53:
54:  214:=28,69,1                           *doppelte Schriftbreite
55:  215:=15                                *Schmalschrift
56:  216:=28,86,1                           *doppelte Schrifthöhe
57:  217:=27,32,6                           *gesperrte Schrift
58:  213:=18, 28,69,0, 28,86,0, 27,32,0 *normal: Aufhebung aller vier
                                                  Kommandos
59:
60:  218:=27,83,0                           *Hochstellen (Superscript)
61:  219:=27,83,1                           *Tiefstellen (Subscript)
62:  220:=27,84                             *normal: Aufhebung der beiden
                                                  Kommandos
63:
64:  205:=27,45,1                           *Unterstreichen ein
65:  206:=27,71,27,69                       *Fett+Schattiert ein
66:  207:=27,52                             *Kursiv ein
67:  208:=27,45,0, 27,53, 27,70, 27,72      *normal: Aufhebung aller drei
                                                  Kommandos
68:
69:  203:=27,97,1,13                        *Zeile zentriert drucken
70:  204:=27,97,2,13                        *Zeile rechtsbündig drucken
```

Listing der Installation des Druckertreibers

auch aus den Steuerzeichen vor und hinter der Überschrift in Abb. 2 ersichtlich.

Ein eigener Zeichensatz

Um die vielen Steuerzeichen einigermaßen übersichtlich im Text zu integrieren, ist ein eigener Zeichensatz sehr von Vorteil. Wer sich nicht die Mühe machen will, selbst einen zu entwerfen, kann das kleine GFA-BASIC-Programm verwenden. Die resultierenden Zeichen sind mit ASCII-Code in Abb. 3 angegeben.

Ein Tip für eigene Zeichensätze: Obwohl es naheliegt, für die Steuerzeichen eine inverse Darstellung zu wählen (damit sie sich besser vom restlichen Text abheben), hat sich dies als unpraktisch erwiesen. Es wird nämlich praktisch unmöglich, die aktuelle Cursor-Position festzustellen, weil der nichtblinkende Cursor von den Steuerzeichen nicht unterscheidbar ist.

Die Symbolik ist im wesentlichen einleuchtend. Eine Ausnahme stellen möglicherweise die Zeichen für Schmalschrift, doppelte Zeichenhöhe,.. dar. Hierbei sollen die kleinen Kästchen Buchstaben symbolisieren. Normalschrift wird durch zwei Kästchen dargestellt, schmale Schrift durch drei schmale, breite Schrift dagegen durch nur ein breites Kästchen. Bei gesperrter Schrift ist der Abstand der beiden Kästchen deutlich größer, bei doppelter Schrifthöhe sind die Kästchen höher dargestellt.

Erweiterungsmöglichkeiten, Alternativen

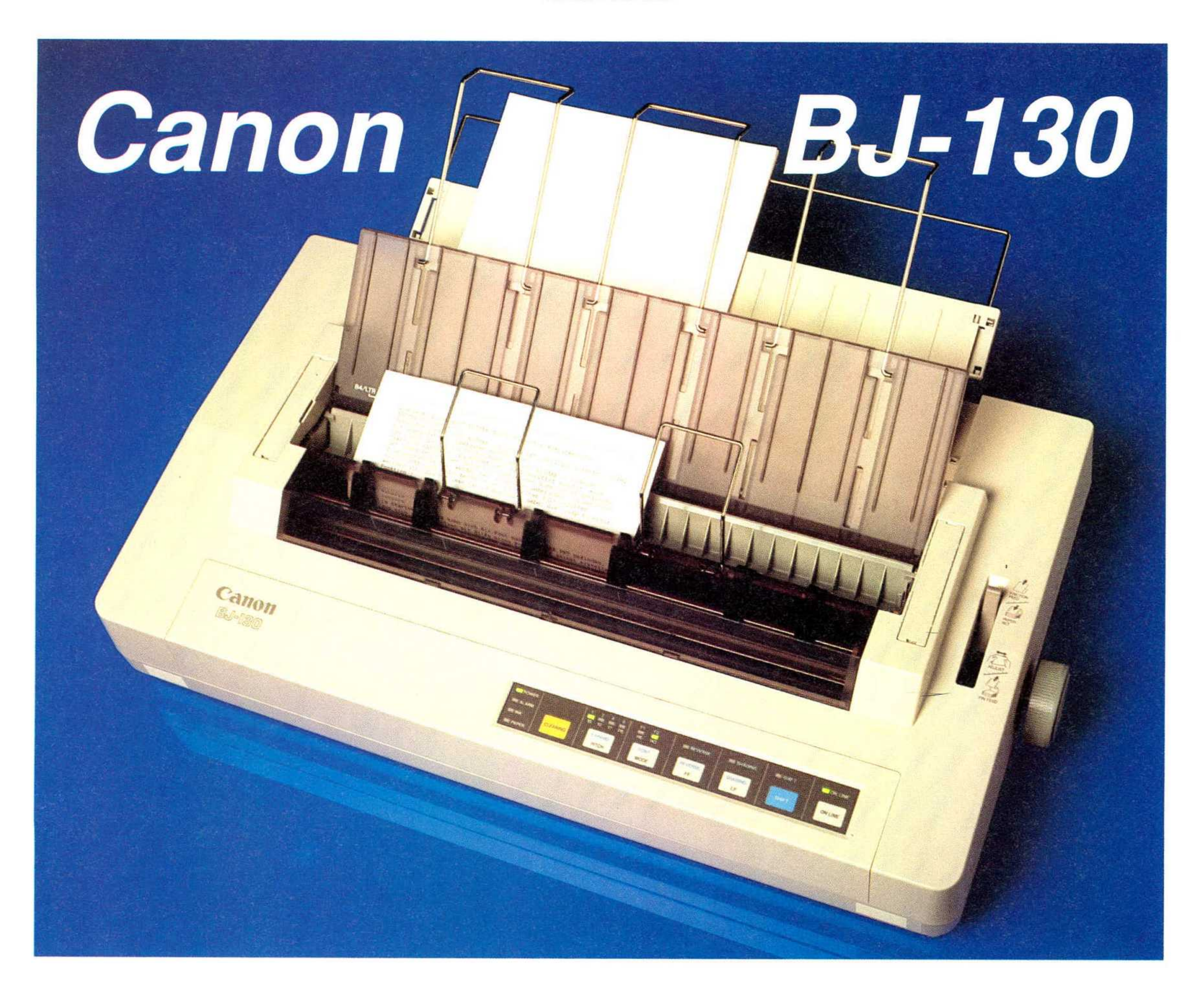
Besitzer von Druckern mit mehreren Schrifttypen (z.B. Courier, Orator, Script, Roman,..) können Codes zur Wahl einer dieser Schriften integrieren. Wer einen farbtüchtigen Drucker besitzt (und zudem bereit ist, die teueren Farbbänder für Textausdrucke zu verwenden), kann durch geschickte Farbwahl den Ausdruck übersichtlicher und effektvoller gestalten.

Der hier gezeigte Weg zur Druckerinstallation ist natürlich nicht der einzig mögliche. Wer auf Kopf- und Fußzeilen nicht verzichten will, muß dafür einige andere Einschränkungen in Kauf nehmen. Wer einen Drucker ohne das leistungsstarke Blocksatzkommando des NEC-P6 besitzt, muß sich mit dem von TEMPUS vorgegebenen Blocksatz zufriedengeben - oder sich etwas Besseres einfallen lassen. Die hier vorgestellten Ideen sind auch nicht vom Himmel gefallen - etwas Phantasie und sehr viel Geduld beim Probieren (nach der Methode "Versuch und Irrtum") sind notwendig.

Michael Kofler

```
 1:  ' einige Zeichen des 8*16 Zeichensatzes ändern und den neuen
 2:  ' Zeichensatz als .FNT-Datei für Tempus speichern
 3:  ' GFA-BASIC 2.0, nur SW
 4:  ' Michael Kofler, Mai 89
 5:  ' (c) MAXON Computer GmbH, Eschborn
 6:  '
 7:  DIM font%(1024) !Speicher f.d. neuen Zeichensatz
 8:  font%=VARPTR(font%(0))    !dessen Startadresse
 9:  OPEN "o",#1,"vid:"        !alter Zeichensatz wird
                                (mit Sonderzeichen
10:  adr%=XBIOS(2)             ! über den Datenkanal VID:
11:  FOR i%=0 TO 3*64 STEP 64 ! am Bildschirm
                                ausgegeben und von
12:    CLR a$                  ! dort in den
                                Zeichensatzspeicher
13:    FOR j%=0 TO 63          ! gelesen
14:      a$=a$+CHR$(i%+j%)     ! adr zeigt auf die
                                Startadresse
15:    NEXT j%                 ! des Bildschirms
16:    PRINT AT(1,1);
17:    PRINT #1,a$;
18:    FOR j%=0 TO 63
19:      FOR k%=0 TO 15
20:        POKE font%+(i%+j%)*16+k%,PEEK(adr%+j%+k%*80)
21:      NEXT k%
22:    NEXT j%
23:  NEXT i%
24:  DO                        ! anschließend werden einige der
25:    READ asc%               ! Zeichen entsprechend den Werten
26:    EXIT IF asc%=-1         ! aus den DATA-Zeilen geändert
27:    adr%=font%+asc%*16
28:    FOR i%=0 TO 15
29:      READ bitmuster%
30:      POKE adr%+i%,bitmuster%
31:    NEXT i%                 ! zum Schluß wird der neue
32:  LOOP                      ! Zeichensatz gespeichert
33:  BSAVE "font_16.fnt",font%,4096
34:  '
35:  DATA 13,0,0,0,0,2,2,2,2,2,18,50,126,48,16,0,0
36:  DATA 157,0,0,60,4,60,32,60,0,24,0,0,126,126,0,0,0
37:  DATA 192,0,0,28,20,20,20,28,0,0,8,12,126,12,8,0,0
38:  DATA 193,0,0,28,16,28,4,28,0,0,8,12,126,12,8,0,0
39:  DATA 194,0,0,92,84,84,84,92,0,0,8,12,126,12,8,0,0
40:  DATA 195,0,0,46,42,42,42,46,0,0,16,48,126,48,16,0,0
41:  DATA 196,0,0,28,16,28,4,28,0,0,16,48,126,48,16,0,0
42:  DATA 197,0,0,28,20,20,20,28,0,0,16,48,126,48,16,0,0
43:  DATA 198,0,46,42,42,42,46,0,8,28,62,8,8,8,62,28,8
44:  DATA 199,0,60,32,60,4,60,0,8,28,62,8,8,8,62,28,8
45:  DATA 200,0,60,32,60,36,60,0,8,28,62,8,8,8,62,28,8
46:  DATA 201,0,60,36,60,36,60,0,8,28,62,8,8,8,62,28,8
47:  DATA 202,0,46,34,46,40,46,0,8,28,62,8,8,8,62,28,8
48:  DATA 203,0,0,0,8,8,73,107,127,107,73,8,8,0,85,0,0
49:  DATA 204,0,0,0,1,1,17,25,253,25,17,1,0,0,85,0,0
50:  DATA 205,0,0,0,0,102,102,102,102,102,102,102,62,0,126,126,0
51:  DATA 206,0,31,63,60,60,255,255,60,60,60,60,60,0,85,0,0
52:  DATA 207,0,24,25,51,62,60,120,120,108,204,198,195,0,85,0,0
53:  DATA 208,0,0,0,92,126,102,102,102,102,102,102,102,0,85,0,0
54:  DATA 209,0,126,90,126,76,74,0,126,90,94,88,120,0,85,0,0
55:  DATA 210,0,0,0,0,0,0,46,42,42,42,46,0,0,85,0,0
56:  DATA 211,0,0,0,0,0,0,46,34,46,40,46,0,0,85,0,0
57:  DATA 212,0,0,0,0,0,0,46,40,46,34,46,0,0,85,0,0
58:  DATA 213,0,0,0,0,0,108,108,108,108,108,108,108,0,85,0,0
59:  DATA 214,0,0,0,0,0,60,60,60,60,60,60,60,0,85,0,0
60:  DATA 215,0,0,0,0,0,84,84,84,84,84,84,84,0,85,0,0
61:  DATA 216,0,108,108,108,108,108,108,108,108,108,108,108,0,85,0,0
62:  DATA 217,0,0,0,0,0,99,99,99,99,99,99,99,0,85,0,0
63:  DATA 218,0,24,60,126,24,24,24,0,85,0,0,0,0,0,0,0
64:  DATA 219,0,0,0,0,0,0,0,0,85,0,24,24,126,60,24,0
65:  DATA 220,0,24,24,24,126,60,24,0,85,0,24,60,126,24,24,24
66:  DATA -1
```

Programm zur Erstellung eines Zeichensatzes mit den Steuerzeichen



Es muß nicht immer Laser sein

Zettel für die Pinwand drucken Sie nur noch mit Signum!? Briefe ans Finanzamt sind mindestens mit Calamus' Vektorfonts zu erstellen? Für Sie ist ein 24-Nadler mit Superhochauflösungsgrafik eigentlich nur Provisorium?

Dann sind Sie ein Perfektionist.

Sie meinen, Ihrer Vorstellung vom ultimativen Schriftbild kommt nur ein Laserdrucker nahe?

Dann sind Sie - mit Verlaub - auf dem Holzweg.

Fortan brauchen Sie sich nicht mehr mit ominösen Publishern oder anderen Zauberprogrammen herumzuschlagen. Ihre Textverarbeitung genügt, um mit Selbstgedrucktem Ohs und Ahs zu ernten. Und dazu braucht's keinen Laser-, sondern nur den Tintenstrahl. Ganz nebenbei geht der Druck mit dem heutigen Testgerät, einem Canon Bubble Jet BJ-130, auch noch superleise vonstatten.

Der Bubble Jet ist ein Tintenstrahldrucker, der aus 48 Düsen das Papier beschreibt. Dabei wird die Tinte in den Düsen erwärmt, so daß sich eine Blase bildet, die abreißt und sich auf den Flug in Richtung Papier macht. (Mehr darüber in [1]). Die Tintentechnik hat den entscheidenden Vorteil, daß kein mechanischer Kontakt mit dem Papier entsteht und somit auch kein Geräusch. Der BJ-130 ist so leise, daß man bequem während des Drucks telefonieren kann, ohne daß der Partner am anderen Ende der Leitung etwas davon bemerkt. Einzig der Papiervorschub macht sich akustisch bemerkbar. Gleichermaßen ist der mechanische Verschleiß weitaus geringer. Der Aufbau der Kopfbewegungsmechanik ist daher beim Bubble Jet nicht so aufwendig wie bei anderen Druckern dieser Leistungsklasse.

Keine Technik ohne Nachteile - auch der Tintenstrahl hat seine. Es sind nämlich logischerweise keine Durchschläge möglich. Zum Drucken von Frachtbriefen fällt er also aus. Vor allem beim Grafikdruck erinnert man sich an die vergangene ABC-Schützenzeit und daran, daß man

bereits damals ewig Probleme mit den verordneten Füllern hatte. Denn wenn viele schwarze Flächen gedruckt werden sollen, trocknet die Tinte langsam und die Gefahr des Verwischens ist hoch. Doch auch im trockenen Zustand läßt sie sich wieder auflösen - mit Wasser. Der Druck ist demnach nicht dokumentenecht.

Aus der Nähe...

... betrachtet wirkt er sachlich schlicht, der Bubble Jet. Helles Gehäuse, farblich abgesetztes Tastenfeld, dunkle Einzelblattklappe. Doch wo ist der Traktor, wo geht hier eigentlich das Papier hinein? Ein Blick auf die Rückseite des Geräts verrät: Hier arbeitet ein automatischer Einzug für beschnittene Blätter. Der ist also serienmäßig, den Aufsatz-Zugtraktor für das endlos perforierte Druckgut gibt's als

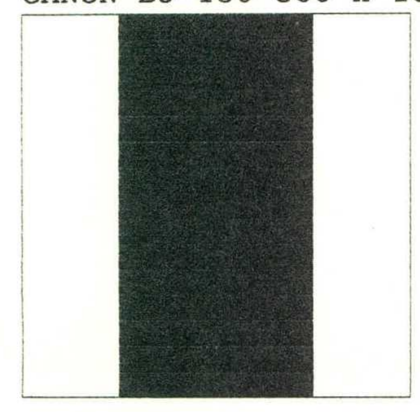
CANON BJ-130
Hoch auf dem gelben Wagen
Hoch auf dem gelben Wagen sitz
Hoch auf dem gelben Wagen

Schriftprobe

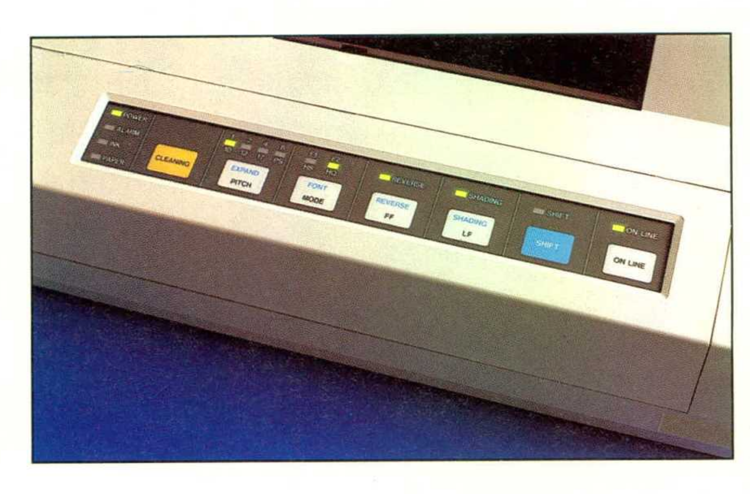
Sonderzubehör. Wo wir bereits von hinten schauen: Auch die serielle Schnittstelle (RS 232-C) gibt's gegen Aufpreis.
Nun wird das Papier eingelegt - kein Problem. Das ist beim Bubble Jet genauso einfach wie bei einem Fotokopierer. Einschalten, der Netzschalter sitzt verrenkungsanfällig seitlich links. Los geht's. Vorher allerdings zeigt der Canon all seine Lämpchen, fährt den Kopf hin und her, reinigt ihn und zeigt sich dann betriebsbereit. Ein rotes Signal zeigt Papiermangel, besinnt sich jedoch eines Besseren, nachdem der automatische Einzug nach dem ersten Datenempfang Papier gefunden hat.

Murphy schlägt mit aller Gewalt zu. Auf

CANON BJ-130 360 x 180 DPI



Grafikprobe



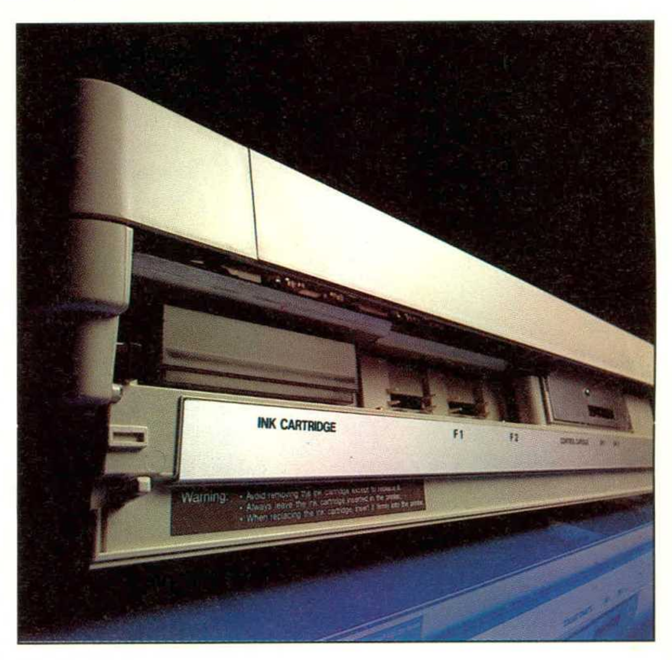
Das Tastenfeld des Bubblejet

dem Papier ist nur Gwrksl - das war ein Satz mit X. Der Grund ist schnell gefunden. Das Zauberwort heißt mal wieder Emulation und der Bubble Jet ist auf IBM Proprinter trainiert. Das ist zwar sehr schön, aber all meine Programme erwarten einen Epson oder - das wär noch besser - einen NEC P7. Und genau das ist kein Problem. Als Sonderzubehör gibts ein EPROM nebst Handbuch und schon hält sich der BJ-130 für eben diesen 24-Nadler.

Jetzt geht's also endlich los mit den ersten Probedrucken, deren Qualität beeindruckt. Selbst geübte Augen tippen bei so kontrastreicher Schrift und gleichmäßigen Schwärzen auf einen Laserdrucker. Auf den Abbildungen finden Sie Text- und Grafikproben. Da der Tintendruck sehr scharfe Ränder hinterläßt, ergibt sich ein solch feines Druckbild. Beim Nadeldrucker hingegen drücken die Nadeln ja ein Textilband gegen das Papier, was unweigerlich zu unscharfen Rändern des Abdrucks führt.

Werfen wir noch einen Blick auf das Tastenfeld des Canon. Neben Einstellungen für die Schriftbreite kann hier auch die Schriftgröße selbst festgelegt werden. Die Zeichen werden jedoch lediglich vergrößert, bei sechsfachem Zoom zeigen sich denn auch deutlich die 'Treppen'. Ferner lassen sich hier einige Stile, der Zeichensatz und Zeilen-, sowie Seitenvorschub wählen. Drei Lampen warnen vor offener Klappe, fehlendem Papier und Tintenmangel. Eine Taste ist mit 'Cleaning' beschriftet und erweckt die Neugier des Testers. Sollte der Bubble Jet gar etwas katzenhaftes haben und sich selbst putzen können? Nach dem Druck der Taste fährt die Maschine den Kopf in die Ruheposition ganz links (was immer passiert, wenn keine Daten zum Druck anliegen) und spült die Düsen im Kopf offensichtlich durch. Tatsächlich deuteten einmal während des Tests feine waagerechte weiße Streifen auf eine verstopfte Düse hin. Nach Aktivierung der Selbstreinigung war das Problem beseitigt.

Nach Öffnen der Klappe an der Vorderseite des Gerätes zeigt sich der Tintenbehälter links, daneben die Schächte für die Fontmodule, die zusätzlich erhältlich sind (Zur Verfügung stehen Orator, Gothic und Gothic Outline). Auch die DIP-Schalter haben hier einen gut erreichbaren Platz gefunden.



Die Tastatur

Im Betrieb...

... zeigt sich der Canon Bubble Jet als ein treuer Gesell, den man gern mit Daten füttert, da er sein Werk ja so flüsterleise verrichtet. Verschiedene Tests ließen

keinerlei Probleme auftreten. Mit der P7-Emulation läßt er sich von allen Programmen, die diesen Drucker ansprechen, nutzen. Unser Druckertestprogramm bescheinigt ihm volle Kompatibilität. Beim Blättern durch das Handbuch fallen zwei Stile auf, die der BJ-130 aufgrund seiner Tintenstrahltechnik zuläßt: Schattiert und Invertiert (siehe Bild). Ein Hauch von 48-Nadel - Verzeihung: Düsen - Gefühl kommt auf, wenn man sich die Befehle für die Download-Fonts ansieht. Obwohl der Bubble Jet über eben so viele Düsen verfügt, arbeitet er ja wie ein normaler 24-Nadler. Er spricht also als Ersatz für eine Nadel immer zwei Düsen an, die auch entsprechend enger beieinanderliegen. Doch intern arbeitet er sehr wohl mit 48 verschiedenen Düsen, was auch die hohe

Druckertest schattiert
Druckertest invertiert

Der Tintenstrahl macht's möglich

Qualität des Schriftbildes erklärt: Die Fonts sind mit einer Auflösung von 360 mal 360 Punkten pro Zoll definiert. NEC hatte einmal für die P6 Plus-Modelle Fontkarten angekündigt, die es erlauben sollten, mit dieser Auflösung 'normale' Schrift zu drucken. Beim Canon Bubble Jet gehört das dazu. Und freundlicherweise erlaubt es ein bisher unbekannter Steuercode, eigene Zeichensätze mit dieser Auflösung zu definieren. Wer viel Geduld hat, kann die Signum!-Zeichensätze umformatieren, als Downloads zum Drucker schicken und daraufhin mit jeder Textverarbeitung in Signum!-Qualität drucken. Nochmal die Extrema zum Vergleich: Mit der Draft-Qualität eines 9-Nadlers hat man die Wahl zwischen 117 Punkten, aus denen man ein Zeichen aufbauen kann. Beim Bubble Jet sind es für jedes einzelne Zeichen genau 1728 Punkte.

Die Geschwindigkeitstests trüben das Gesamtbild allerdings stark. Die Tabelle zeigt den Bubble Jet im Vergleich zu anderen (Nadel-) Druckern, die für ähnliche - professionelle - Anwendungen konzipiert wurden. Unser Kandidat ist demnach ziemlich langsam. Das zeigt sich bereits bei dem Test 'DIN 32751', bei dem ein einseitiger Geschäftsbrief gedruckt wird. Der Vorsprung der nadeligen Kollegen wächst dann natürlich mit der Dokumentlänge. Aber auch bei der Grafik erweist sich der BJ-130 als behäbig. Die

Dokumentlänge Text: 33396 Bytes
Dokumentlänge Graphik: 32643 Bytes
Genannt: Zeit zur Datenabnahme / Gesamtzeit für Druck

Drucker	LQ-Einzel	LQ-Endlos	Draft-Endlos	Graphik	DIN 32751
Canon BJ-130	2:42/16:58	-	-	0:23/0:24	37 s
Epson LQ-2500	-	7:47/10:04	3:49/4:56	0:16/0:17	34 s
NEC P6 PLUS	-	0:18/9:10	0:18/5:46	0:16/0:20	34 s
OKI ML 393C	0:50/7:46	0:39/6:27	0:20/3:02	0:16/0:28	26 s

Datenabnahme im Textmodus ist verhältnismäßig schnell. Hier zeigen sich die 32 kByte Speicher, die Canon seinem Bubble Jet gegönnt hat.
Nun ist man versucht, die Tintentechnik als langsam bloßzustellen, doch interessanterweise ist gerade der Druck selbst schnell. Der Bubble Jet hat jedoch einen sehr langsamen Zeilenvorschub. Auch das Einziehen eines Blattes mit dem Einzug dauert ewig lang - da ist man von anderen Herstellern mehr Tempo gewohnt. Und, wie gesagt, wertvolle Zeit vertrödelt die Maschine beim Vorschub des Papiers von einer Zeile zur nächsten. Schade eigentlich, richtig fixer Druck hätte dem Bubble Jet gut zu Gesicht gestanden und seine Bestimmung als Bürodrucker weiter unterstrichen.

Grafik...

... mit dem Canon Bubble Jet zu drucken ist ein verlockender Gedanke. Verlieren doch die Nadeldrucker gerade hier - bei schwarzen Flächen - Terrain und erweisen sich nicht als der Qualität letzter Schluß. Außerdem erzeugen sie beim oft langwierigen Grafikdruck ein geradezu nervtötend sägendes Geräusch. Der BJ-130 verspricht, dem ein Ende zu setzen, zumal er eben über die Auflösung der NEC-Drucker von 360 mal 360 Punkte pro Zoll verfügt. Und tatsächlich werden Bilder mit sehr guter Qualität zu Papier gebracht. Die Grafikprobe zeigt gute Schwärzen, geringe helle Streifen (die durch ungenauen Zeilenvorschub zustande kommen) und sehr saubere senkrechte Linien. Die Maßhaltigkeit ist nicht so gut, in der senkrechten läßt sich eine Abweichung von ca. einem Prozent ausmachen.

Einer der DIP-Schalter läßt es zu, die Tintenmenge, die für den Grafikbetrieb genutzt wird, zu erhöhen. Das ist nötig, wenn eine reine Strichzeichnung, die z.B. mit einem CAD-Programm erstellt worden ist, zu Papier gebracht werden soll. Das Druckbild ist dann aber immer noch recht dünn. So richtig problematisch wird der Grafikdruck erst, wenn die schwarzen Flächen groß werden. Dann beginnt die Tinte auch schon mal vom Papier zu fließen, während sich das Blatt noch im Drucker befindet. Dagegen hilft, die Intensität wieder zurückzunehmen. Man schaltet dann ewig mit dem DIP-Schalter hin und her.
Mit Signum! gibt es leider in diesem Zusammenhang die größten Schwierigkeiten. Seine Präzision erweist sich für den BJ-130 als Bumerang. Denn die P24-Zeichensätze von Signum! sind in Ihrer Intensität natürlich auf die normalen 24-Nadeldrucker abgestimmt. Die meisten Zeichensätze hinterlassen mit dem Bubble Jet denn auch ein viel zu mageres Druckbild. Sie sehen im Bild den vielbe-

Canon BJ-130 / 360 DPI
Dieser Test mit Signum!
zeigt die Qualität des
Tintendrucks.

Der Bubblejet und Signum!

nutzten Zeichensatz Antikro, der mit einer Auflösung von 360 mal 360 Punkte pro Zoll gedruckt wurde.

Unterm Strich...

... zeigt sich der Bubble Jet als prädestiniert für den Arbeitsplatz der Sekretärin. Ein Korrespondenzdrucker schlechthin ist er. Die Bedienung ist einfach und narrensicher, während des gesamten Testes wurde nicht ein einziges Blatt schief eingezogen oder gar zerknüllt. Mit den 64 Zeilen, die er auf ein DIN A4-Blatt druckt, sind die Ränder oben und unten klein genug, um vorhandenes Briefpapier korrekt zu bedrucken. Die Qualität im Text-

modus ist äußerst gut, wem die Standardschrift Courier nicht gefällt, der kann ja auf eines der Fontmodule ausweichen. Von intensivem Grafikdruck mit dem BJ-130 ist aus oben angeführten Gründen allerdings abzuraten.

Was den Bubble Jet am geeignetsten erscheinen läßt, ist eben das Fehlen der Geräusche. Betrachtet man derzeit Computerarbeitsplätze in Firmen, Büros oder Praxen, dann hat man den Eindruck, wir befänden uns in der Steinzeit. Von Ergonomie am Arbeitsplatz kann so gut wie nie die Rede sein. Da stehen Tastaturen viel zu hoch, spiegeln sich Lichtermeere in unscharfen Monitoren, erfordert die Bedienung der Geräte oft unmögliche Verrenkungen und - lärmen die Drucker. Da wird eben nur zu deutlich, daß technischer Fortschritt immer zuerst eine Verbesserung der Maschinen bedeutet, während der sie benutzende Mensch und seine Belange oft genung in den Hintergrund geraten. Unter diesem Aspekt ist ein leiser Drucker wie der Bubble Jet als Investition in die Ergonomie eines Arbeitsplatz zu werten. Denn, bei aller Begeisterung für die Sache als solche, wer kann denn einen klaren Gedanken fassen, wenn neben ihm ein Nadeldrucker fünfzig Briefe druckt?

Jetzt könnte eingewandt werden, daß ja auch Laserdrucker sehr leise sind. Das ist zwar richtig, allerdings gehen von ihnen durch das ständigen 'Einbrennen' des Toners Gase aus, über deren Verträglichkeit noch nicht allzu viel bekannt ist. Subjektiv riechen sie aber nicht gut und sind daher auch nicht sonderlich angenehm. Fairerweise muß man die Preise für eine gedruckte Seite vergleichen: Bei Laserdruckern rechnet man mit Kosten um 12 Pfennig pro Blatt. Dieser Preis ergibt sich vor allem aus den hohen Kosten für den Toner. Leider macht Canon den Vorteil des Bubble Jet durch einen hohen Preis für die Tinte wieder zunichte. Laut Hersteller soll eine Farbkartusche für den Bubble Jet gut für mindestens eine Million Zeichen sein. Das sind - je nach Art der Dokumente - um 700 Seiten. Pro Seite ergeben sich also Kosten von 11 Pfennigen zuzüglich des Papierpreises. Verwendet man normales Kopierpapier, mit dem der BJ-130 die besten Ergebnisse erzielt, so kostet eine Seite insgesamt ca. 13 Pfennig. Damit ist der Tintenstrahl genauso teuer wie ein Laser und mancher wird, wenn man den Geschwindigkeitsvorteil betrachtet, doch letzteren vorziehen.

Die Wahl des Papiers ist außerordentlich wichtig, wenn man mit dem Bubble Jet ein sauberes Druckbild erzeugen will. Ist das Papier zu glatt und die Oberfläche zu geschlossen, so trocknet die Tinte sehr schlecht und verwischt leicht. Ist hingegen die Oberfläche rauh oder das Papier zu holzig, so wird die Tinte unregelmäßig aufgesogen. Der Druck sieht dann aus, als schriebe man mit einem Füller auf Löschpapier. Die besten Ergebnisse erzielt man mit feinem Fotokopierpapier, das nebenbei auch noch erschwinglich ist.

Vergeblich sucht man am Bubble Jet den Hebel zur Druckkopfverstellung. Bei Nadeldruckern dient er dazu, den Kopfabstand an unterschiedliche Papierdicken anzupassen. Interessanterweise bedruckt der Tintenstrahler trotzdem klaglos Aufkleber oder Briefumschläge. Die Entfernung von Kopf zu Papier scheint also nicht so entscheidend zu sein wie bei den Nadlern. Einzige Bedingungen an das Druckgut sind eben, daß es der automatische Einzug erfassen kann (wenn man nicht den Stacheltraktor hat) und daß die Oberfläche geeignet ist.

Der BJ-130 ist also ein rundherum empfehlenswertes Gerät, wenn in der Hauptsache Text gedruckt wird. Wer ein Herz für seine Mitarbeiter, Kollegen, Nachbarn, etc. hat, der sollte ihn in die engere Wahl ziehen. Sein Anschaffungspreis ist mit 3135 DM zwar nicht ganz billig, aber er ist durchaus sein Geld wert. Wenn er auch kein Meister der Geschwindigkeit ist, so tötet er wenigstens nicht noch den letzten Nerv. Außerdem belohnt das exzellente Druckbild für längeres Warten.

IB

[1] Gut gedruckt ist halb geschrieben, ST-Computer 4/89, S. 40 ff.

Preise (können je nach Händler bis zu 15% variieren)

Gerät:	*3135,- DM*
Zugtraktor:	*256,- DM*
Serielle Schnittstelle:	*296,- DM*
Emulations-EPROM:	*203,- DM*
Fontmodul:	*336,- DM*
Tintenkartusche:	*80,- DM*

Garantie: 1 Jahr auf alle Teile

Canon Bubble Jet BJ-130

+ Sehr gutes Schriftbild
+ Sehr leise
+ Problemlose Bedienung

- Langsam
- Grafik problematisch
- Tinte sehr teuer

MAMOS

Ein neuer Modula-Konkurrent

Mammuts sind laut Fremdwörter-DUDEN eine "ausgestorbene Elefantenart mit langhaarigem Pelz der Eiszeit mit 5m langen Stoßzähnen". Ob das Mammut Modula-2 System MAMOS aus der Schweiz eine Totgeburt aus vergangenen Zeiten oder ein Programmierpaket mit neuer Stoßkraft ist, soll dieser Test überprüfen.

MAMOS wird auf zwei Disketten und mit einem 250-seitigen Handbuch geliefert. Momentan muß das Paket noch direkt bei Mammut-Software in der Schweiz für DM 249,- bestellt werden.

Installation

Zur Installation brauchen lediglich alle Dateien auf die Festplatte kopiert zu 0werden. Bei Verwendung eines Diskettensystems reicht eine Kopie der Systemdiskette, die als Arbeitsdiskette weiterverwendet wird.

Eine spezielle Installationsdatei (Bild 1) setzt die Parameter, die das System und die Shell steuern. Diese Installationsdateien können übrigens geschachtelt werden, so daß man eine Datei mit speziellen Parametern für ein Projekt halten kann, das eine weitere mit allgemeinen Einstellungen nachlädt.

Wie andere Modula-Implementierungen stützt sich diese Konfigurierung auf eine Reihe von Variablen, die das System erkennt. Neben der Einstellung der Suchpfade für Dateien in PATH können z.B. in COMPOPT die Compiler-Optionen voreingestellt werden.

Der Shell lassen sich über TOOL weitere Hilfsprogramms ins Menü schreiben; die in KEEP verzeichneten Systemprogramme hält die Shell resident, so daß das Compiler-Modul nur einmalig vom Massenspeicher geladen werden muß.

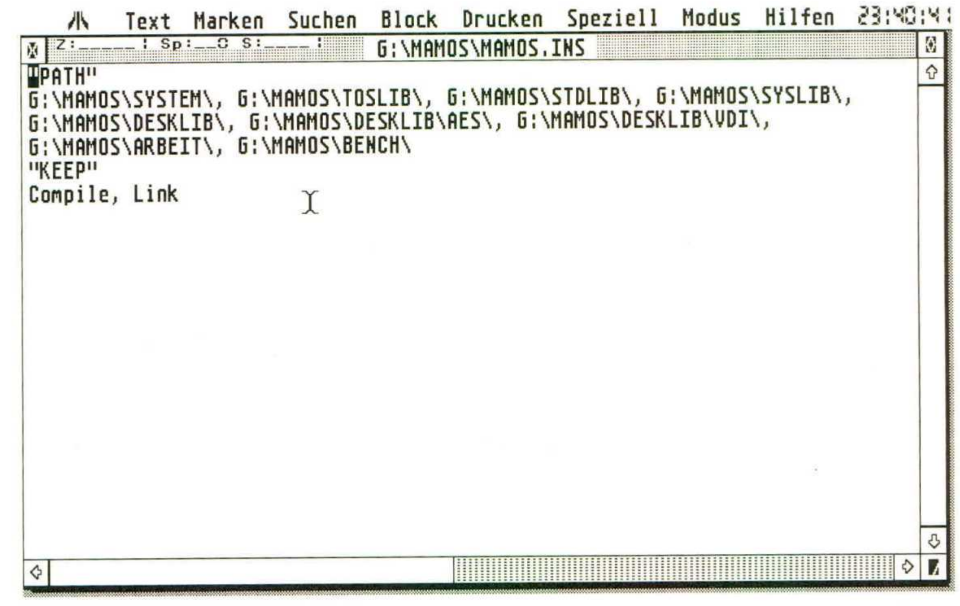


Bild 1: Eine spezielle Installationsdatei setzt die Parameter, die das System und die Shell steuern.

Die Installation ist einfach und birgt keinerlei Fallstricke. Die Variable TOOL wird in der vorliegenden Version leider noch nicht beachtet.

Die Shell

Die Shell ist recht einfach gehalten und erinnert stark an LPR-Modula (Bild 2). Die Menüs "Arbeit" und "Wahl" rufen die einzelnen Systemprogramme auf, entweder auf die Hauptdatei bezogen oder nach Auswahl per Fileselector.

Unter "Werkzeug" können weitere Objektmodule in die Shell eingebunden werden. Bei einer Neukonfiguration kann automatisch der Editor mit der Installationsdatei aufgerufen werden. Die Shell führt nach Verlassen des Editors erneut einen Installationsdurchgang aus, so daß sie nicht neu gestartet werden muß, wenn z.B. die Suchpfade zu ändern sind.

Insgesamt ist die Shell recht simpel und entspricht eher der Minimal-Shell von LPR, als daß sie mit den komplexeren Oberflächen von SPC oder Megamax mithalten könnte.

Editor

Als Editor verwendet MAMOS eine speziell angepaßte Lizenzversion von TEMPUS 1.1. Die Anpassungen sind in Bild 3 sichtbar: Die Funktionstasten sind mit Modula-Schlüsselwörtern belegt, und auch die Dateiauswahl verwendet die bei MAMOS vorkommenden Datei-Extensionen. (Übrigens sieht man hier auch, daß TEMPUS nur bis Laufwerk G: per Maus gesteuert werden kann.)

TEMPUS ist für seine Geschwindigkeit bekannt und für viele Anwender der Traum-Editor für Programme. Im Vergleich mit der Konkurrenz bietet MAMOS damit einen Editor, der den Neuentwicklungen der anderen Systeme mindestens ebenbürtig ist. Mich persönlich stört, daß TEMPUS nicht in der Lage ist, zwei Zeilen zu verbinden. Ein versehent-

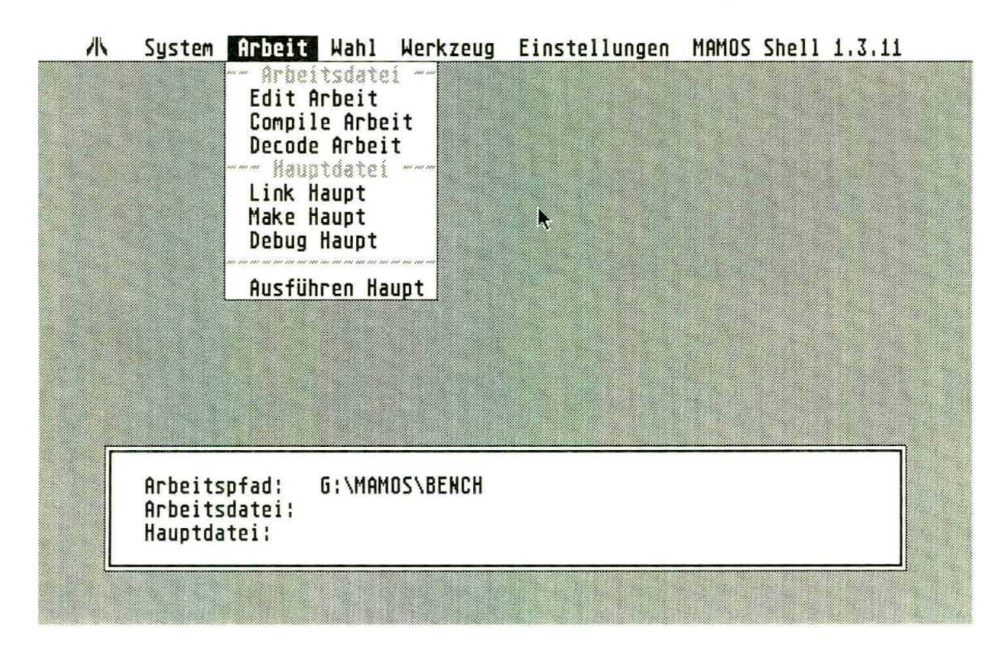


Bild 2: Die Shell von MAMOS-Modula 2

liches Drücken von <Return> verlangt somit die Benutzung der Blockoperationen. MAMOS-Anwender erhalten übrigens beim Kauf von TEMPUS 2.0 bei CCD einen Preisnachlaß von 35 Prozent.

Compiler

Der 1-Pass-Compiler ist eine Portierung des bekannten ETH-Systems. Damit ist er sehr wahrscheinlich fehlerarm und implementiert den vollen Sprachumfang. Die Compileroptionen umfassen die üblichen Schalter z.B. für die Bereichs- und Überlaufüberprüfung. Den Fließkommaroutinen kann per Schalter eine Überprüfung auf NANs (Not A Number) nach IEEE-Spezifikation hinzugefügt werden.

SETs werden mit 16 Bits implementiert. Funktionen müssen Basistypen als Ergebnis haben. Die Daten- und Codegröße ist jeweils auf 32 Kilobyte pro Modul beschränkt, für das gesamte Programm gibt es keine Beschränkungen. Der Compiler arbeitet mit ausreichender Geschwindigkeit und entspricht somit den übrigen ETH-Portierungen.

Benchmarks

Als siebtes Modula-System stellt sich MAMOS den von anderen Tests in der ST-Computer her bekannten Benchmarks. Dazu aber eine Vorbemerkung: Das mitgelieferte sogenannte "maximale" Laufzeitsystem ermöglicht das Arbeiten mit dem Debugger. Ein so gebundenes Programm versucht beim Programmstart, die entsprechenden Module des Debuggers automatisch nachzuladen. Damit eigneten sich die erzeugten Programme nicht mehr zur Zeitmessung unter Guläm. Ich mußte daher auf ein Benchmark-Set zurückgreifen, das für die erste Version von SPC (damals noch ohne Linker) geschrieben wurde.

Die Zeiten sind mit dem 200Hz-Zähler des ST gemessen, indem vor und nach den eigentlichen Bench-Routinen jeweils dessen Stand gelesen und die Differenz als Laufzeit angezeigt wird. Damit hat MAMOS allerdings einen Vorteil, da sämtliche Statements zur Initialisierung des Laufzeitsystems nicht gemessen werden. Die Benchmarktabelle wird sobald wie möglich entsprechend korrigiert, bis dahin muß den MAMOS-Werten eine gewisse Zeitspanne hinzuaddiert werden. Ein Wert von 3 bis 5 Sekunden (wahrscheinlich verursacht z.B. durch die Speicheranforderungen bei GEMDOS) ergab sich bei SPC nach dem Umstieg auf die Guläm-Messungen: ein Wert, der sicher auch in etwa für MAMOS gelten wird. Die Ergebnisse sehen Sie in Bild 4. Es zeigt sich deutlich, daß MAMOS wie SPC und LPR vom Züricher ETH-Compiler von Nikolaus Wirth abstammen. Mit der oben angedeuteten Initialisierungszeit im Hinterkopf liegt die Geschwindigkeit der Compilate im üblichen Rahmen.

Auffällig sind die Zeiten der Tests 27a und 28, die beide auf Library-Module zugreifen. Es scheint, daß hier noch keine Optimierungen vorgenommen wurden und sie noch in reiner Hochsprache formuliert sind.

Benchmark 27 konnte nicht durchgeführt werden, da ein Fehler in der MathLib0 auftrat: Der Versuch, exp(4.2) zu errechnen, endete mit einem arithmetischen Überlauf. Daraus läßt sich auf einen Fehler in der Routine schließen, denn die Konkurrenzsysteme liefern - auch bei unterschiedlich langen REALs - das korrekte Ergebnis 66.686 ohne Klagen ab.

Insgesamt liegt der Compiler also im Standard; unter den Systemen, die vom ETH-Compiler abstammen, lassen sich kaum noch gravierende Unterschiede feststellen.

Bibliotheken

In den Bibliotheken bemüht sich MAMOS um eine möglichst große Kompatibilität zum unter MS-DOS recht verbreiteten Logitech Modula-2 in der Version 3.0. Damit ergeben sich einige Anpassungen, insbesondere bei Filesystems, die teilweise auf spezielle MS-DOS-Features abstellen. Gegenüber Logitech sind die Bibliotheken auch teilweise wiederum an den ST angepaßt, z.B. bei der Behandlung der 68000-Interrupts.

Wieweit die Kompatibilität reicht, konnte leider nicht getestet werden. Der Versuch, sich bei den Bibliotheken an einen existierenden Marktstandard zu halten, ist jedoch erfreulich. Es scheint möglich, Programme mit Textfenstern einfach auf den ST zu portieren. Bei der anstehenden allgemeinen Normierung der Bibliotheken durch ISO und DIN wird MAMOS sich allerdings neu für eine Bibliotheksstruktur entscheiden müssen.

Die Standardbibliothek umfaßt zusätzliche Module für dynamische Zeichenketten, schnelle Byte-orientierte Operationen und lange SETs. Neben den bekannten AES- und VDI-Modulen sind "höhere" zur Event- und Fensterverwaltung vorhanden. GraphicsWindows und TextWindows lehnen sich an die Vorschläge von Wirth und die Logitech-Routinen an. Bei Textfenstern ist die Verwendung von Texteffekten einschließlich Invertierung möglich.

Die Bibliotheken sind in jeder Hinsicht ausreichend und komplett. Durch die Kompatibilität zum Logitech MS-DOS-System ist eine hohe Portabilität gegeben. Der bei den Benchmarks aufgetretene Fehler läßt allerdings vermuten, daß sich noch kleinere Bugs in den Modulen verstecken.

Linker

Der Linker arbeitet im Moment noch nicht optimierend, eine Erweiterung, die allerdings geplant ist. Sehr komfortabel

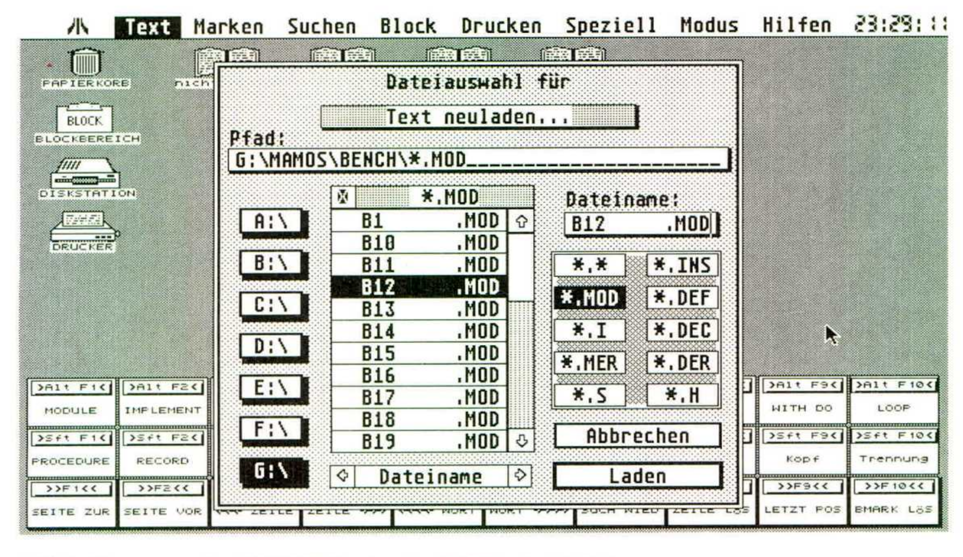


Bild 3: Eine angepaßte TEMPUS-Version ist als Editor beinhaltet.

ist die Auswahl des gewünschten Laufzeitsystems. Man hat die Auswahl zwischen "Maximal" und "Minimal", was den Debugger-Aufruf an- und abschaltet. Für die Behandlung von Fließkommazahlen gibt es drei Möglichkeiten: normale Software-Routinen für die Arithmetik, Einsatz des Coprozessors 68881 oder Weglassen der Fließkommaroutinen. Überhaupt wird der Coprozessor intelligent unterstützt. Beim Erstellen eines Stand-Alone-Programms mit gewählter 68881-Option sucht der Linker zunächst nach einem Modul mit der Endung .FPU, das gegebenenfalls auch Coprozessor-Opcodes enthält. Ist ein solches nicht vorhanden, wird das übliche .OBM-Modul gelinkt. Damit lassen sich sehr einfach spezielle Programmversionen für Systeme mit 68881 erstellen, ohne am Quelltext zu manipulieren.

Allerdings haben die geschilderten Möglichkeiten im vorliegenden Paket, Version 1.3, noch einen großen Haken: Sie sind noch nicht vollständig implementiert. Konkret heißt das, daß lediglich die maximalen Laufzeitsysteme mit und ohne 68881-Unterstützung vorhanden sind. Es fehlen alle minimalen Laufzeitsysteme und das Rumpfsystem ohne Fließkommazahlenunterstützung.

Die geplante Realisierung des Linkers ist - insbesondere, was die Coprozessorunterstützung angeht - recht leistungsfähig. Ohne Optimierung und angesichts der unvollständigen Implementierung kann er gegenüber den Konkurrenzprodukten jedoch noch nicht überzeugen.

Handbuch

Das 250seitige deutschsprachige Handbuch ist bei MAMOS leider nicht besonders gut geraten. Es macht den Eindruck, mit heißer Nadel gestrickt zu sein, mal werden Umlaute benutzt, mal steht ein simples "Oeffnen". Einige Rechtschreibfehler lassen darauf schließen, daß keine Korrektur erfolgte; an einer Stelle sind Werte sogar handschriftlich nachgetragen worden.

Doch der wichtigste Nachteil ist, daß das Handbuch nicht komplett ist. Nach einer bestimmten Einarbeitungszeit benötigt man beim Programmieren übersichtliche und einfach zugängliche Listings der mitgelieferten externen Module. Ein solcher Anhang macht bei anderen Modula-Systemen oftmals den Großteil des Handbuchs aus. Nicht so bei MAMOS: Im Handbuch findet sich nur eine kurze Beschreibung der Funktionalität der Module, die praktisch wertlos ist (Beispiel zum Modul ResourceHandler: "Operationen, um Resourcen zu laden.").

Vor der wirklichen Arbeit mit MAMOS muß man also zum Druckerschalter greifen und dann mühselig die fast 160 Seiten der Definitionsmodule von Diskette ausdrucken. Als Ergebnis hat man einen Stapel im DIN A4-Format, der umständlich zu handhaben ist. Unbedingt muß ein Anhang mit einem Abdruck der Definitionsmodule in das Handbuch, sein momentanes Fehlen schränkt die Brauchbarkeit extrem ein.

RT Modula-2 Benchmarks

		Jefferson		Softwave		LPR	MAMOS	
Nr.	TDI V3.0		Megamax		SPC V1.41		V1.3	testet ...
1	0:07	0:07	0:07	0:04	0:05	0:05	0:02	Prozeduraufruf
2	1:42	1:33	2:59	1:35	1:31	1:31	1:27	Addition
3	1:21	1:18	1:58	1:20	1:16	1:16	1:12	Increment
4	1:47	1:38	2:59	1:40	1:36	1:36	1:32	Additionsoptimierung
5	1:27	1:23	2:08	1:25	1:21	1:22	1:17	Increment als Vergleich
6	2:09	1:57	3:48	2:01	1:55	1:55	1:52	INTEGER-Addition
7	2:09	1:57	3:48	2:01	1:55	1:55	1:52	CARDINAL-Addition
8	1:11	1:18	1:42	1:05	1:15	1:16	1:12	FOR-Schleife
9	1:21	1:02	1:42	1:05	1:00	1:00	0:56	REPEAT-Schleife
10	1:21	1:18	1:57	1:20	1:15	1:15	1:12	WHILE-Schleife
11	1:04	0:54	1:15	0:38	0:51	0:53	0:48	INTEGER-Parameter
12	1:04	0:54	1:17	0:38	0:51	0:53	0:48	INTEGER VAR-Parameter
13	1:06	0:59	2:19	0:33	0:57	0:57	0:53	RECORD-Parameter
14	0:34	0:30	0:41	0:20	0:28	0:28	0:24	RECORD VAR-Parameter
15	0:49	0:49	1:33	0:42	0:46	0:47	0:43	Konstanten-Optimierung
16	0:51	0:51	1:33	0:44	0:49	0:49	0:45	Konstanten-Optimierung
17	1:28	1:26	2:06	1:19	1:23	1:24	1:20	Expression-Optimierung
18	1:42	1:22	1:59	1:15	1:19	1:20	1:16	Expression-Optimierung
19	0:37	0:36	0:55	0:30	0:33	0:34	0:29	Zwischenergebnis-Optimierung
20	0:37	0:35	0:55	0:30	0:33	0:34	0:29	Zwischenergebnis-Optimierung
21	0:09	0:11	0:13	0:09	0:08	0:09	0:05	IF-Statement
22	0:13	0:13	0:16	0:11	0:11	0:12	0:07	IF durch CASE ausgedrückt
23	0:38	0:33	0:41	0:28	0:30	0:31	0:27	CASE-Statement
24	0:40	0:39	1:03	0:38	0:36	0:36	0:33	CASE durch IF ausgedrückt
25	0:47	1:03	—	2:09	0:49	0:48	0:42	REAL-Arithmetik
26	2:05	—	1:32	2:18	2:07	2:00	1:58	LONGREAL-Arithmetik
27	1:52	5:42	—	—	3:51	3:16	—	REAL-Library
27a	5:39	—	2:35	35:40	3:33	—	16:13	LONGREAL-Library
28	1:21	1:21	0:40	0:30	0:51	—	2:20	String-Library
29	2:10	2:07	2:13	1:48	1:44	2:04	2:01	ARRAY-Zugriffe
30	0:09	0:10	0:17	0:19	0:08	0:08	0:04	RECORD-Zugriffe

Alle Zeiten mit time-Kommando von Guläm gemessen
MAMOS 1.3 mit 200Hz System-Timer gemessen
Meßgenauigkeit bis zu ±0.5 Sekunden
Angabe „—": Sinnlos bzw. keine Bibliotheken

Bild 4: Die Benchmarks

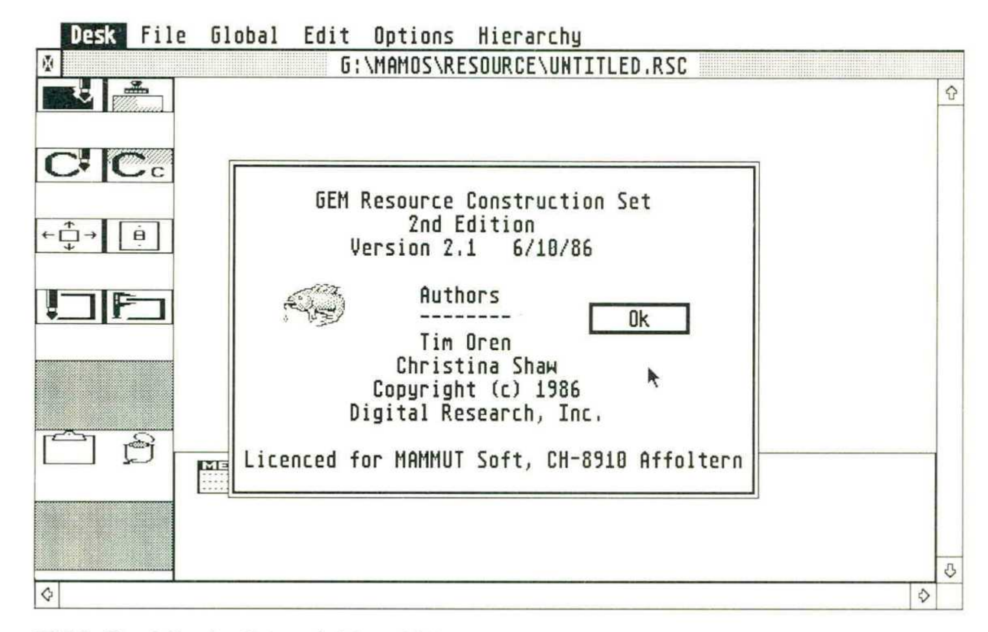


Bild 5: Ebenfalls mitgeliefert wird das RCS 2.1 von Digital Research

Besonderheiten

Dem Paket liegt erfreulicherweise ein Resource-Construction-Set bei; es handelt sich um das RCS von Digital Research in der Version 2.1 (Bild 5). Gegenüber der berüchtigten, bombenträchtigen ersten Version dieses Programms sind die Fehler beseitigt worden, und man hat ein leistungsstarkes RCS vorliegen, das im Komfort allerdings vom Kuma-RCS übertroffen wird.

Weiterhin sind ein Decoder zum Disassemblieren von Modulen und ein Inliner zum Einbinden von vorgefertigten Assembler-Routinen vorhanden. Ein Ordner enthält ein paar Public Domain-Quellen mit eher informatikträchtigen Themen wie Balanced-Trees oder Fibonacci-Zahlen.

Ein Make-Programm ist zwar angekündigt und auch im Handbuch schon beschrieben, fehlte in dem Testexemplar aber noch. In der nächsten Version soll es nachgeliefert werden.

Fazit

MAMOS ist in der vorliegenden Version 1.3 noch nicht komplett. Es fehlen einige der beworbenen Features, das Handbuch muß dringend den angesprochenen Anhang erhalten, und auch die Libraries scheinen noch einige Optimierungen nötig zu haben. Wer das spezielle Problem der Übertragung von Quellen unter MS-DOS mit Logitech-Modula hat, ist mit MAMOS gut bedient.

Dennoch: Auch wenn das mitgelieferte DR-RCS und TEMPUS das Paket aufwerten, kann es die Konkurrenz in der jetzigen Form nicht ausstechen. Wahrscheinlich wird sich in den nächsten Versionen noch einiges ändern, momentan aber gibt es bessere Modula-Systeme.

RT

Bezugsadresse:

Mammut-Software
Condrau & Brunner
Pfruendhofstr. 23
CH-8910 Assoltern/Albis

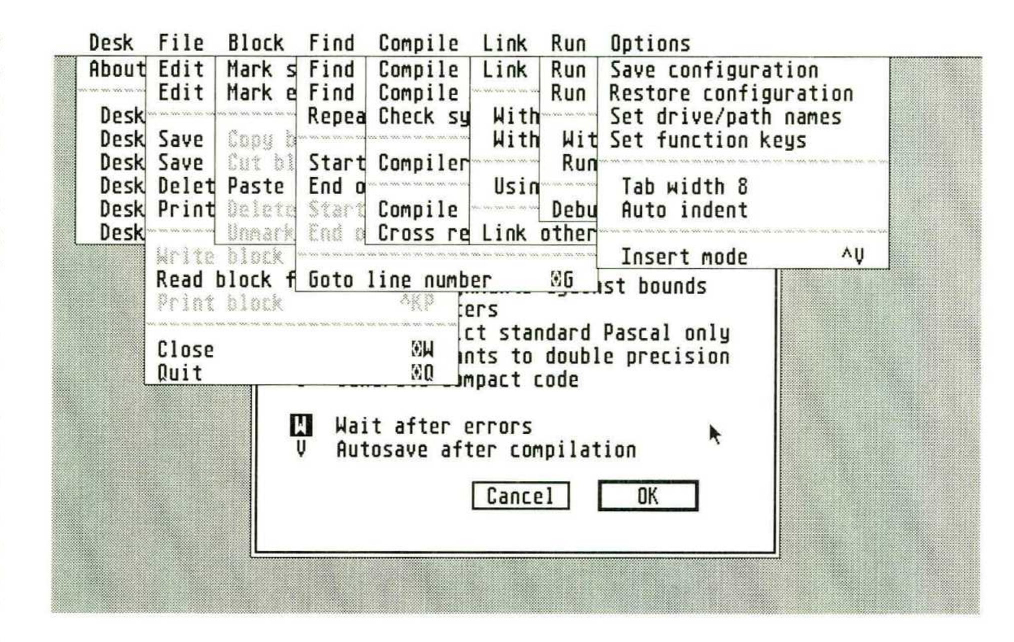
Schritt zurück zum Fortschritt

Neue Version 2.15 von Prospero Fortran und Pascal

Seit vielen Jahren gibt es nun schon die Programmiersysteme von Prospero für den Atari ST, neben Pascal und Fortran neuerdings auch C. Nach dem großen Sprung von den recht unkomfortabel zu bedienenden Versionen 1.x zu den GEM-eingebundenen Anfang 1988 vergrößerte sich die Schar der Anwender erheblich. Mit diesen kamen jedoch auch immer neue Verbesserungsvorschläge.

Seit kurzem nun liefert Prospero die Version 2.15 von Pro Fortran bzw. von Pro Pascal for GEM aus, in der auf einige dieser Vorschläge eingegangen wurde. Nachdem zu beiden Systemen bereits ausführliche Testberichte erschienen sind (ST Computer 2/88 und 4/88), soll hier außer einigen ergänzenden Bemerkungen nur auf die seither eingebrachten Neuerungen eingegangen werden. Vorweggenommen sei, daß der angekündigte Ein-Pass-Compiler wie auch die Spracherweiterungen (etwa DOUBLE COMPLEX oder DO-WHILE-Schleifen in Fortran) leider noch nicht verwirklicht wurden.

Die auffälligste Veränderung besteht darin, daß Compiler und Linker nun wieder in der Kommandozeilenversion mitgeliefert werden. Dieser Schritt zurück geschieht wohl vor allem auf die Klagen der vielen Tempus-verwöhnten ATARI-Besitzer hin, die schon lange darunter gelitten haben, daß die Benutzeroberfläche von Prospero ("Werkbank") die



Verwendung eines eigenen Editors nicht zuläßt. Die Werkbank gibt es natürlich weiterhin, mit einer geeigneten Shell (wie etwa der GULAM-Shell aus dem PD-Bereich) ist es aber jetzt auch möglich, einen schnellen Editor mit den Programmiersystemen von Prospero zu kombinieren.

Wohlgemerkt: Eine solche Shell ist nicht im Lieferumfang enthalten. "Prosperianer" der ersten Stunde können jedoch ihre alte wieder hervorholen, die schon vor Jahren die ersten Prospero-Compiler gesteuert hat, und diese fast unverändert wieder einsetzen, da seitdem nur wenige Parameter hinzugekommen sind. Die genaue Syntax wird ebenso wie die Unterschiede zu alten Versionen in einer READ.ME-Datei auf Diskette kurz erläutert.

Die meisten Anwender werden vermutlich weiterhin mit der mitgelieferten Werkbank arbeiten, die trotz aller Mängel einen recht hohen Komfort bietet. Einer der wesentlichen Fehler ist mittlerweile beseitigt: Durch das Konzept des integrierten Editors wurde das Abspeichern des gerade bearbeiteten Quelltextes auf Diskette/Platte nur bei der Beendigung der Arbeit automatisch durchgeführt - in den Stunden der Entwicklungszeit zuvor mußte der Programmierer schon selbst

daran denken, sonst machten Systemabstürze und wackelige Netzstecker Hunderte mühsam eingetippter Programmzeilen zunichte. Jetzt gibt es endlich eine Option, die NACH jedem erfolgreichen Compilerlauf für ein automatisches Speichern sorgt; dabei achtet die Werkbank sogar darauf, daß der Quelltext eine frühere Systemzeit erhält als die Binärdatei, um zukünftige Make-Utilities zu unterstützen.

Bemerkt der Compiler einen Syntaxfehler, kann wie gewohnt abgebrochen werden; hier funktioniert das Anspringen der fehlerhaften Stelle im Text nun ziemlich zuverlässig. Mittlerweile arbeitet die Werkbank auch in der niedrigen Auflösung, hierfür ist eine eigene Menüzeile vorgesehen. Die Cursorposition (Zeile, Spalte) wird jetzt ständig am oberen Fensterrand angegeben. Schwierigkeiten gibt es immer noch mit dem deutschen Tastaturlayout; eine Syntaxprüfung wird hierzulande nicht über Alt-Y, sondern über Alt-Z ausgelöst. Ein bisher undokumentiertes neues Feature: Der Menüpunkt "Compile and Link" läßt sich nun alternativ über die Tastenkombination Alt-J anwählen.

Aus der neuen Werkbank heraus können Blockbereiche und ganze Dateien ausgedruckt werden, allerdings ohne jeglichen Komfort - weder ein Formatieren von Quelltexten noch eine Druckeranpassung sind vorgesehen.

Ein wenig schneller ist das System auch geworden, besonders die Ladezeiten von Texten und Programmen und das Scrollen im Text haben sich beschleunigt. Insbesondere der Linker profitiert vom schnelleren Einlesen der Bibliotheken, nur - o Schreck - ausgerechnet der Compiler ist sogar noch etwas langsamer als zuvor! Hier wird sich wohl erst in den zukünftigen Ein-Pass-Versionen etwas ändern.

Geändert hat sich leider nichts an den Laufzeitfehlermeldungen: Angaben über die Nummer der Zeile, in der der Fehler aufgetreten ist, beziehen sich immer auf die Gesamtzeilenzahl, Include-Dateien werden mitgerechnet. Um hiermit Fehler aufstöbern zu können, muß somit etwa bei einem GEM-Programm die Länge von GEMCONST.PAS, AESCOMM.PAS, und wie sie alle heißen, bekannt sein und abgezogen werden.

Die Standardbibliotheken wurden leicht modifiziert, um sie zum neuen C-Compiler kompatibel zu machen. Die Prospero-Systeme unterstützen ja mehrsprachiges Programmieren, was den Vorteil hat, etwa den Großteil eines Programms in Pascal schreiben zu können, Berechnungen im Komplexen in Fortran auszuführen und zeitkritische Teile C zu überlassen. Hierzu mußten auch der Linker und die Bibliotheksverwaltung abgeändert werden, was sich für den Normalanwender jedoch nicht merklich auswirkt - Groß- und Kleinschreibung in externen Symbolen sind jetzt äquivalent, der Linker unterstützt "load-time relocation". Allerdings muß ein bestehendes Programm für die Bearbeitung mit dem ebenfalls modifizierten Debugger neu gelinkt werden.

Eine Anpassung an das neue TOS 1.4 ist nicht erfolgt, in der GEM-Bibliothek fehlen weiterhin die AES-Funktionen *Fsel_ExInput* und *Wind_New*. Hier ist CCD mit seiner aktuellen Pascal-Version 2.06 schon einen Schritt weiter, ebenso wie bei der Unterstützung der Prozessoren 68010 und 68020. Während dort über eine einfache Compileroption entsprechender Code generiert wird, kann der 68020 bei Prospero nur zusammen mit einem Floating-Point-Koprozessor und einer Zusatzbibliothek für etwa DM 180.- programmiert werden. Immerhin gibt es keine Schwierigkeiten mit TOS 1.4, selbst der Source-Level-Debugger arbeitet weiterhin problemlos.

Zusammenfassend ist zu sagen, daß sich ein Update (Kostenpunkt ca. DM 30.-) zur Zeit vorwiegend für diejenigen lohnt, die ihre Prospero-Compiler mit einem anderen Editor einsetzen wollen. Hierzu ist jedoch zunächst die Anpassung einer geeigneten Shell notwendig, da eine solche ebensowenig wie ein Make-Utility oder ein Resource Construction Set mitgeliefert wird. Es ist aber immerhin eine Utility-Diskette mit all diesen Dingen inklusive Quelldateien der GEM-Bibliotheken angekündigt, laut Kundeninfo für den "more serious programmer" gedacht. Bleibt zu hoffen, daß die Wartezeit auf den nächsten großen Schritt, zum schnellen Fortran-Ein-Pass-Compiler, nicht zu lange dauern wird.

Andreas Hill

Bezugsadresse:

EDV-Beratung
Friedrich Plünnecke
Hinterm Dorfe 21
3325 Lengede
Tel.: 05174/1637

Pro Pascal for GEM			**V 2.11**	**V 2.15**
Laden	30 k	<	14 s	8 s
Speichern	30 k		8 s	8 s
Blättern (Seiten)	30 k		31 s	30 s
Scrollen (Zeilen)	30 k		70 s	58 s
Syntaxcheck	30+50 k		45 s	43 s
Compiler	30+50 k		107 s	115 s
Linker	16+105 k		40 s	36 s
Größe der Binärdatei.			15956	15956
Größe der Programmdatei			40960	40960

Alle Werte auf Festplatte und garantiert handgestoppt.
Die Ausführungszeiten fertiger Programme sind identisch.

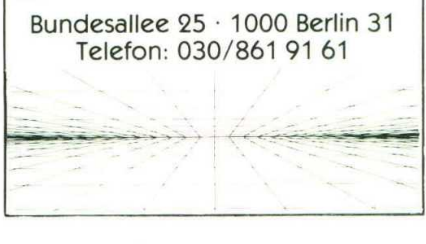
Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

1000 Berlin



DATAPLAY
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61





alpha computers g.m.b.h.
u. a. alphatronic, atari, commodore, dai, epson, sord mit pips, nec
hard-/software nach maß — servicetechnik
Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/891 1082

Computare
Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d





ATARI ST Hardware Spezialist
Computertechnik
ST COMPUTER-PD's vorrätig!
Beratung und Vorführung von Hard-und Software
1000 Berlin 65 · Pankstr. 42
Tel. 030/465 70 28-29

1000 Berlin



ATARI
. . . wir machen Spitzentechnologie preiswert.
Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

Bei uns werben bringt
GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

COMPUTER-STUDIO
Schlichting
. . . die etwas andere Computerei
Ihr Spezialist in Berlin für Hardware + Zubehör Eigenes Softwarestudio über 1000 verschiedene Titel am Lager
ATARI-Fachmarkt
NEC-Fachhandel · MS-DOS Fachmarkt
Katzbachstraße 6 + 8 · 1000 Berlin 61
☎ 030/7864340

2000 Hamburg

Computer & Zubehör-Shop
Gerhard u. Bernd Waller GbR
Kieler Straße 623
2000 Hamburg 54
☎ 040/570 60 07
BTX 040 570 52 75

Hardware
Software
Beratung
Service



ATARI Systemfachhändler
Munsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400
Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 0 40/641 50 91

RADIX Bürotechnik
Heinrich-Barth-Straße 13
2000 Hamburg 13
Telefon (0 40) 44 16 95
NEU: Software Shop

GMA mbH
040/25 12 41/6 7 17
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

Computer
Hardware · Software · Zubehör



2000 Norderstedt



2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt

Heiligengeiststr. 20, 2120 Lüneburg
Tel. 04131 / 46122, Btx 402422
Mo.-Fr. 9°°-18°° und Sa. 9°°-13°°

2210 Itzehoe



Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 04 31 / 56 70 42

2800 Bremen



Faulenstraße 48—52
2800 Bremen 1
Telefon (04 21) 17 05 77

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler

Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

3000 Hannover



Großer Hillen 6
3000 Hannover 71
0511 - 52 27 11

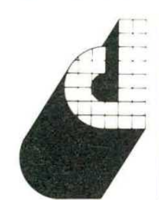
3000 Hannover



trendDATA Computer GmbH
Am Marstall 18-22 · 3000 Hannover 1
Telefon (05 11) 1 66 05-0

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36



DATALOGIC
COMPUTERSYSTEME
ATARI ST- COMPUTER · HARDWARE · SOFTWARE
BERATUNG · SERVICE · VERKAUF
CALENBERGER STR. 26
3000 HANNOVER 1
TEL.: 0511 - 32 64 89

3040 Soltau

F & T Computervertrieb

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91 / 1 65 22

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software

Woltorfer Str. 8, 3150 Peine
Tel. 0 51 71 / 60 52/3 o. 0 51 73 / 79 09

3170 Gifhorn

C OMPUTER
H AUS
G IFHORN
INHABER AXEL RITZ

D-3170 GIFHORN
POMMERNRING 38
TELEFON (05371) 5 44 98

MITGLIED DER COMTEAM
DIE COMPUTER-PARTNER
IHR FACHHÄNDLER FÜR ATARI, AMSTRAD, AEG, LEO, NEC, OKI, EPSON

3400 Göttingen

Büroeinrichtungs-Zentrum



3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51 / 38 57-0

3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (05 61) 70 00 00

4000 Düsseldorf

HOCO
EDV ANLAGEN GMBH
Ellerstraße 155
4000 Düsseldorf 1
Telefon 02 11 / 78 52 13

Hard und Software
Werner Wohlfahrtstätter

Atari	**Ladenlokal**
Public Domain	Irenenstraße 76c
Atari Spiele	4000 Düsseldorf-Unterrath
Atari Anwender	Telefon (02 11) 42 98 76

BERNSHAUS GmbH
Bürotechnik – Bürobedarf

Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

4010 Hilden

Beachten Sie unsere Anzeige in diesem Heft!



Computer · Drucker · Software · Bücher · Service
Gustav-Mahler-Straße 42-44
Tel. (0 21 03) 3 18 80 + 4 12 26

Bei uns werben bringt
GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

BUF

4130 Moers



- Service-Center
- ATARI Fachhändler
- Hardware
- Software
- Erweiterungen

COP Computer Service GmbH
Essenberger Straße 2H · 4130 Moers
Telefon (0 28 41) 2 35 85

4300 Essen

ATARI Systemfachhändler

PROFI COMPUTER STUDIO

KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4320 Hattingen

Ihre Tür zur Zukunft:
KARSTADT computer-center
hardware · software · problemlösungen
Hattingen, Große Weilstr. 18-20, Telefon (0 23 24) 2 09 73

4330 Mülheim

BNS Computer und Bürotechnik Vertriebsgesellschaft mbH
Dickswall 79 4330 Mülheim Telefon 02 08/3 40 34

Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene

Tandon Computer · VICTOR · SEL-Fernkopierer · NEC · ATARI · OKI

4422 Ahaus

ATARI · Epson · Fujitsu Molecular · NCR · Tandon · Schneider · Star

OCB

OCB-Computershop
Wallstraße 3
4422 Ahaus
Tel. 0 25 61/50 21

OCB-Hard- und Software
Wessumerstraße 49
4422 Ahaus
Tel. 0 25 61/50 21

4430 Steinfurt

CBS GmbH
COMPUTERSYSTEME
Tecklenburger Str. 27
4430 Steinfurt-Burgsteinfurt
☎ 02551/2555

Bei uns werben bringt
GEWINN
Sprechen Sie mit uns.
Heim Verlag ☎ 0 61 51 / 5 60 57
BUF

4500 Osnabrück

Heinicke-Electronic

Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99

Wir liefern Micro-Computer seit 1978

4520 Melle

CBS GmbH
COMPUTERSYSTEME
4430 Steinfurt Tel. 02551/2555
Haferstraße 25 4520 Melle
Tel.: 05422/44788

4600 Dortmund

ATARI SYSTEM-Fachhändler



BÜRO STUDIO BOLZ

4600 Dortmund 1 · Brauhausstraße 4
Telefon (02 31) 52 77 13 - 16

Elektronik
Computer
Fachliteratur

ATARI-System-Fachhändler

4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84

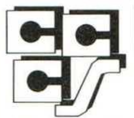


ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Kampstraße 1 · 4600 Dortmund
Telefon (02 31) 5 43 91

cc Computer Studio GmbH



Atari-Systemfachhändler
PCs von Tandy Schneider Peacock
Drucker von Star Brother NEC

Elisabethstr. 5
4600 Dortmund 1
Tel. 0231/528184 Tx 822631 cccsd Fax 0231/528131

4650 Gelsenkirchen-Horst

MENTIS GmbH

Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

4700 Hamm

computer center

Moderne Büroausstattung · Büromaschinen · Büroorganisation · Büromöbel · Werkstatt/Service

RÜTER · BÜRO RÜTER

Heinrich Rüter GmbH & Co. KG
4700 Hamm 1
Gustav-Heinemann-Straße 19/21
Telefon 02381/1404-0

4708 Kamen



D+M Computer
Weststraße 45
4708 KAMEN
02307 - 17052

Die jungen Spezialisten für PC

BTX *0230715181
Telefax 0 23 07 / 1 83 67

4712 Werne

Vogler & Trümper
Hard- und Software



Lünener Straße 14
4712 Werne
Tel. (0 23 89) 5 14 95

4800 Bielefeld

hardware software organisation service **CSF**

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

Carl-Severing-Str. 190
4800 Bielefeld 14

Telefon: 05 21 / 45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21 / 45 99-123

MICROTEC
Software
Hardware
Beratung
Service

4950 Minden

Computer Vertriebs GmbH PCM

- Software
- Hardware
- Organisation
- Beratung
- Schulung

Obermarktstr. 21
4950 Minden
0571 - 2 14 48

5000 Köln



AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
TEL. (02 21) 21 91 71

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71 / 26 24

5253 Lindlar



5300 Bonn



Hardware · Software · EDV-Zubehör
Telefon 02 28/22 24 08
COCO GmbH · Schumannstraße 2 · 5300 Bonn 1

In Bonn Ihr Ansprechpartner für
- Hardware
- Software
- Zubehör
- Beratung

ATARI ST — MS-DOS

Computer & Beratung Behnck
(0228) 67 70 21
Bestellungen nehmen wir jederzeit entgegen; oder vereinbaren Sie einen Beratungstermin!

5414 Vallendar



ATARI-Systemfachhändler für Mayen-Koblenz
Ihr autorisierter Fachhändler für GTC- Personalcomputer, Star, Epson und NEC
Wir schreiben BERATUNG und SERVICE groß!
Zentrale: 5414 Vallendar, Rheinstr. 117, TEL. 0261/61727
5419 Dierdorf, Hauptstraße 50
5500 Trier, Ehrangerstr. 31

5500 Trier



Güterstraße 82 · 5500 Trier
☎ 06 51 / 20 97 10
Fordern Sie unsere Zubehör-Liste an!

5600 Wuppertal

COMPUTER FINKE
COMPUTER finke
ATARI - SYSTEMFACHHÄNDLER
KIPDORF 22 · 5600 WUPPERTAL 1 · TEL. 0202 45 32 33
HARDWARE · SOFTWARE · ZUBEHÖR · SERVICE · SCHULUNGEN
ATARI
... wir machen Spitzentechnologie preiswert.

MEGABYTE
Computer Vertriebs GmbH
Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (02 02) 8 19 17

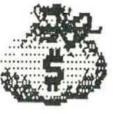
5630 Remscheid

COM SOFT
Nordstraße 57 · 5630 Remscheid
Telefon (0 21 91) 2 10 33

5650 Solingen

MegaTeam
Computer-Vertriebs-OHG
Kölbach – Finke
Hardware – Software – Zubehör – Service
Rathausstraße 1–3 · 5650 Solingen 1
Telefon (02 12) 4 58 88 · Fax (02 12) 4 73 99

Bei uns werben bringt
GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057
BUF

5800 Hagen



Vertragshändler Axel Böckem
Computer + Textsysteme
Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Telefon (0 23 31) 7 34 90

5900 Siegen



Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95

6000 Frankfurt

Eickmann Computer
Der Atari-System-Fachhändler !
z.B.: Festplatten von 30 MB bis 110 MB für Atari ST und Mega ST, Zusatzaufrüstungen für Ihre Festplatten bis 110 MB, Umrüstung Ihres SM 124 in einen EM 124 Multisync für alle Auflösungen, Slotkit für PC 1...
besuchen Sie unser Fachgeschäft:
In der Römerstadt 249
6000 Frankfurt 90-Praunheim
Telefon (069) 763409

WAIZENEGGER
Büroeinrichtungen
Kaiserstraße 41
6000 Frankfurt/Main
Tel. (0 69) 2 73 06 - 0

DAS BÜRO-FACHGESCHÄFT!
Müller & Nemecek GmbH
Kaiserstraße 44
6000 Frankfurt/M.
Tel. (0 69) 23 25 44

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon (0 61 51) 5 60 57

6200 Wiesbaden

COMPUTER TREFF
Computerbedarf, PD und Software für ATARI, AMIGA, PC
Nettelbeckstraße 12
6200 Wiesbaden
Tel. (0 61 21) 40 43 02

6240 Königstein

KFC COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Tel. 0 61 74 - 30 33
Mail-Box 0 61 74 - 53 55

6250 Limburg



6300 Gießen



6400 Fulda

Schneider ATARI Commodore
WEINRICH
BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon (06 61) 4 92 - 0

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing

Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6520 Worms



Computersysteme
GmbH

6520 Worms · Friedrichstraße 22
Telefon 0 62 41 / 67 57 - 58

6700 Ludwigshafen

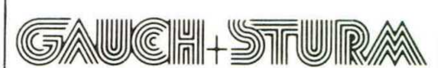
MKV Computermarkt

Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

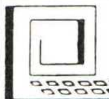
6720 Speyer

THEILLE Computersysteme
Gilgenstraße 4 · 6720 Speyer
Telefon (0 62 32) 772 16

6800 Mannheim



Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912


Computer-Center
am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83 / 84

6900 Heidelberg

JACOM FAMILA-CENTER
Hardware · Software
Schulung · Service

Hertzstraße 1 · 6900 Heidelberg 1
Telefon (0 62 21) 30 24 37

7000 Stuttgart



7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler
ATARI ST
Matrai computer
Matrai Computer GmbH
Bernhauser Str. 6
7022 L.-Echterdingen
☎ **(07 11) 79 70 49**

7030 Böblingen

Verkauf – Service – Software



Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31 / 22 60 15
COMPUTER CENTER

7047 Jettingen

Verkauf – Service – Software

Norbert Hlawinka
Heilbergstraße 3
Im Multi-Center
7047 Jettingen
Telefon (0 74 52) 7 76 15



COMPUTER SHOP

7100 Heilbronn



Bei uns werben bringt

GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057 BUF

Computer-Welt


Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

7150 Backnang




Erfolgreich werben

Sprechen Sie mit uns.
Heim-Verlag ☎ (0 61 51) 5 60 57 BUF

7410 Reutlingen

Werner Brock
Computer-Studio
Untere Gerberstr. 15 · 7410 Reutlingen
Tel. 0 71 21 - 3 42 87
Tx 1 72 414 024 RMI D box rmi taisoft Fax 0 71 21 - 33 97 79
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR
SCHEURER
ATARI COMMODORE CUMANA DATA-BECKER MULTITECH RITEMAN SCHNEIDER THOMSON
7475 Meßstetten 1 · Hauptstraße 10 · 07431/61280

7500 Karlsruhe



MKV GMBH

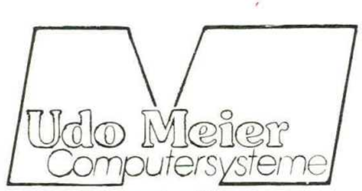
Kriegsstraße 77
7500 Karlsruhe
Telefon (0721) 84613

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk

In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7700 Singen



Ringstraße 4
Telefon (07731) 68222

7730 VS-Schwenningen

BUS BRAUCH & SAUTER
COMPUTER TECHNIK

Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER
computer-fachgeschäft
rösler
Rheingutstr. 1 · ☎ 07531-21832

7800 Freiburg



Kartäuserstraße 59
7800 Freiburg
Telefon: 0761/36870-70
Fax: 0761/25849
PYRAMID COMPUTER GmbH

7850 Lörrach



7890 Waldshut-Tiengen



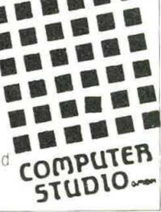
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 07751/3094

7900 Ulm



EDV-Systeme
Software-erstellung
Schulung

Systemhaus:
Frauenstraße 28
7900 Ulm/Donau
Tel. (0731) 28076
Telex 712973 csulm-d



7918 Illertissen

biotech gmbh
technische Informationsysteme
Computerladen

Marktplatz 13
7918 Illertissen
07303/5045

8000 München



COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STRASSE 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

Ihr Spezialist für:
dBMAN - Komplettlösungen
vortex Massenspeicher
Vortex, dBMAN sind eingetragene Warenzeichen zugunsten Dritter
ABAC München
Kellerstraße 11, 8000 München 80
Tel. 089/4489988

Michael Weichselgartner
Ihr Spezialist für Computer
-HARDWARE -SOFTWARE
-BERATUNG -ZUBEHÖR
Chiemgaustraße 152 ▪ 8000 München 90
☎ 089 / 6804642

schulz computer
Schillerstraße 22
8000 München 2
Telefon (089) 597339
Beratung · Verkauf · Kundendienst

8032 Gräfelfing



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464.851043

8170 Bad Tölz

Uwe Langheinrich
Elektronik Center
Wachterstraße 3
8170 Bad Tölz
Telefon (08041) 41565
Bitte Gratisliste anfordern!

8150 Holzkirchen



Service und Beratung sind bei uns inklusive

8330 Eggenfelden



Computer-Centrum
R. Lanfermann

Schellenbruckstraße 6
8330 Eggenfelden
Telefon 08721/6573

Altöttinger Straße 2
8265 Neuötting
Telefon 08671/71610

8400 Regensburg

Zimmermann elektroland

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 0941/95085

8390 Passau
Kohlbruck 2a
☎ 0851/52007

8423 Abensberg

COMPUTERVERSAND WITTICH
Tulpenstr. 16 · 8423 Abensberg
☎ 09443/453

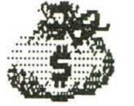


8500 Nürnberg



Bei uns werben bringt

GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

BUF

8500 Nürnberg



8520 Erlangen



Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 09131 / 42076

Zimmermann elektroland

8520 Erlangen
Nürnberger Straße 88
Tel. (09131) 34568

8500 Nürnberg
Hauptmarkt 17
Tel. (0911) 20798

8600 Bamberg



8700 Würzburg



Hardware · Software
Service · Schulung

computer center
am Dominikanerplatz
Ruf (0931) 30808-0

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 09721 / 652154

8900 Augsburg



Unser Plus: Beratung u. Service
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (0821) 528533 oder 528087

Computer Vertriebs- und Software GmbH

ÖSTERREICH

A-1030 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m.b.H

A-1030 Wien
Landstraßer Hauptstraße 2
Hilton-Einkaufspassage

A-1040 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m.b.H

A-1040 Wien · Paniglgasse 18-20
Tel. (0222) 505 78 08, 505 88 93

A-1060 Wien

AMV ATARI SHOP
BÜROMASCHINEN IM GENERALICENTER

Ihr
ATARI Partner

Mariahilferstraße 77-79
(Generalicenter), A-1060 Wien
Tel. (0222) 961951
+
Webgasse 21, A-1060 Wien
Tel. (0222) 5976759

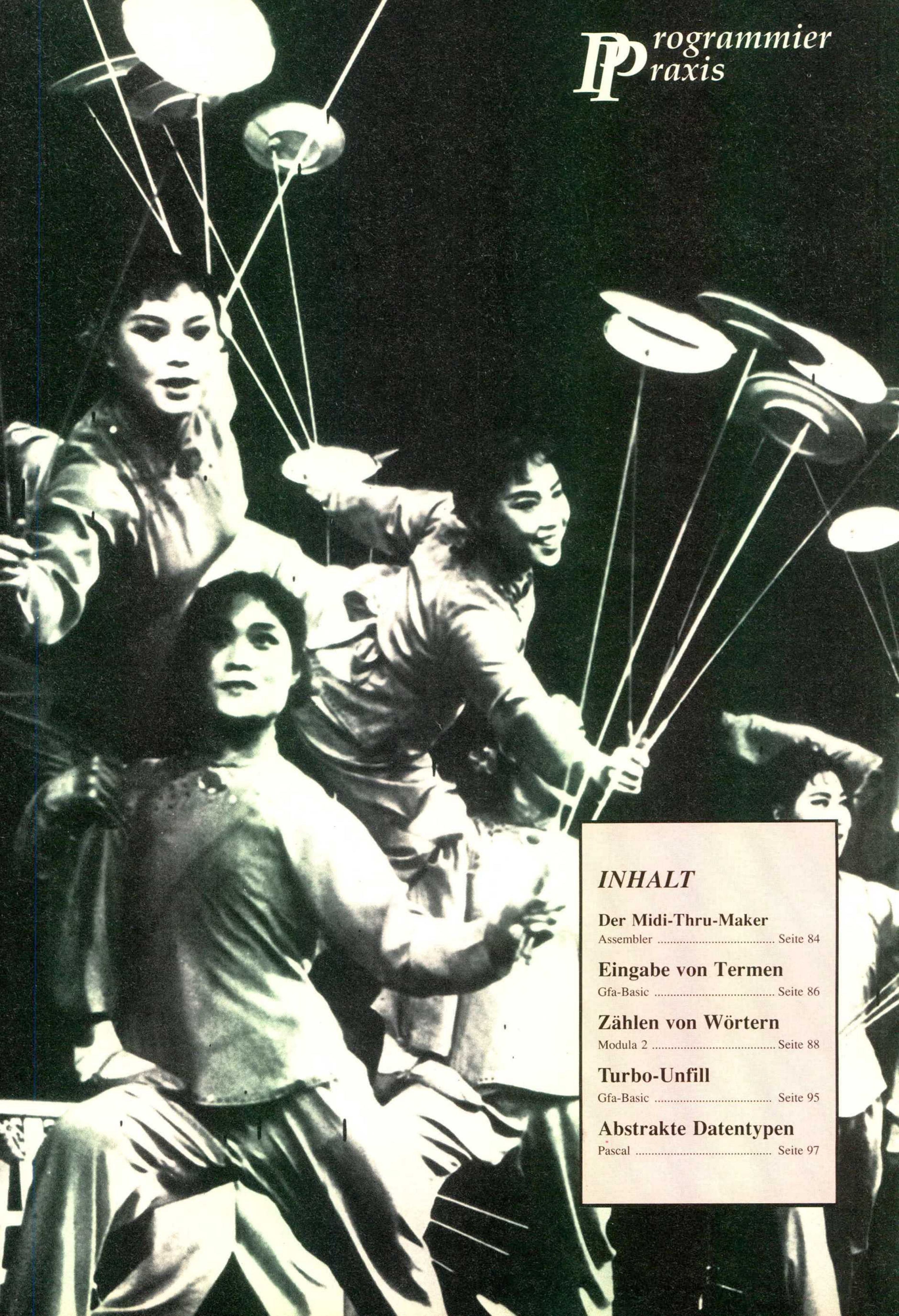
Generalvertretung der Bavaria-Soft

A - 1180 Wien

Ihr ST-Fachhändler in Wien



Tel. (0222) 48 52 56
A-1180 Wien - Schulgasse 63



INHALT

DER MIDI-THRU MAKER

Berthold Dettlaff

UND DA WAR DOCH NOCH DER MUSIKER, DER SICH GERADE MIT DER TEXTVERARBEITUNG HERUMQUÄLTE UND MÜHSAM DEN TEXT ZU SEINEM NEUSTEN HIT IM ZWEI-FINGER-ADLERSYSTEM IN DEN ATARI ST REINTIPPTE, ALS IHN JÄH DIE MUSE KÜSSTE - EIN SUPERLIED SCHWIRRTE IN SEINEM KOPF. ALSO MASTERKEYBOARD AN, SYNTHI NR.1 AN DAS NETZ, EXPANDER UND RHYTHMUS-COMPUTER POWER ON; DANN LÄSST ER SEINEN GEFÜHLEN FREIEN LAUF, DOCH O SCHRECK ... KEIN TON, MIDI GEHT NICHT ...!

Nun kommt Hektik ins Spiel: schnell eine leere Diskette her und den Text abspeichern, raus aus dem Programm, den Software-Sequencer rein und den MIDI-Kanal einstellen. Jetzt endlich kann's losgehen! Doch leider war dabei die Idee auf der Strecke geblieben. Tausendmal war's geschehen, das Tausendunderstemal war genug.

Nie wieder, schwor sich der Musiker und schaffte sich... einen zweiten ATARI an! So geschehen in ???? am ...... Damit niemandem von uns das passiert, kam ich auf Idee: WARUM NICHT MIDI-THRU PER SOFTWARE??? So gesagt, so getan. Aus diesem Gedanken entstand der Midi-Thru-Maker.

Das Prinzip

Der Midi-Thru-Maker arbeitet mit allen Programmen zusammen, die den Tastaturinterrupt nicht verbiegen. Grob passiert folgendes: Ein Teil der Tastaturinterruptroutine wird ersetzt, d.h. es wird nur der MIDI-Abfrageteil ummontiert, alles andere bleibt so, wie's war. Die Adresse der Routine wird dem TOS bekanntgegeben, damit ab sofort bei jedem ACIA Interrupt unsere MIDI-Daten von der MIDI-IN-Schnittstelle gelesen und unverändert nach MIDI-Out geschrieben werden.

Der Midi-Thru-Maker ist also ein nettes kleines Programm, das in 10 Minuten eingegeben ist und sehr Wertvolles für die gestreßten Musiker unter den ATARIANERN leistet.

Das Programm

In den ersten 13 Programmzeilen passiert nichts Weltbewegendes, denn das war alles irgendwo schon einmal da.

Zeile 1-5: Dort wird die Programmlänge berechnet. Die wird nämlich später benötigt, um das Programm resident zu halten.

In den Zeilen 6-9 wird dem TOS die Adresse der Routine mitgeteilt, die den ACIA-Interrupt-Vektor verbiegt; die Ausführung erfolgt natürlich im Supervisor-Modus.

Zeile 11-13: Das Programm verabschiedet sich, verbleibt aber im Speicher, damit der verbogene Vektor nicht ins Leere greift.

Zeile 15-18: Dies ist die "Vektor-Verbiege-Routine". Die Systemvariable bei Speicheradresse $118 beherbergt den ACIA-INTERRUPT-VEKTOR, der für unsere Zwecke ab sofort auf die Routine "MKB" (Zeile 21) zeigen soll. Dann brauchen wir nur noch den Empfänger-Interrupt zu erlauben (sicher ist sicher) - fertig.

Wichtig wird für uns die Funktion ab Zeile 25 (Label 'anfang'). Dort wird im Register A0 die Basisadresse der MIDI-ACIA, nämlich $FFFC04, untergebracht. Der Status der MIDI-ACIA wird in Zeile 26 gelesen. Falls der schwebende Interrupt von der MIDI-Schnittstelle kommt, ist Bit 0 des ACIA-Statusregisters '1'. Wenn nicht, dann rührt der Interrupt von einem Tastatur- oder Mausereignis her; in diesem Falle wird direkt die IKBDSYS-Routine angesprungen. (Beachten Sie bitte, daß je nach TOS-Version diese Adresse von der unten angeführten abweichen kann. Näheres entnehmen Sie der Tabelle am Ende der Erläuterung.) Wurde ein MIDI-Ereignis erkannt, wird das Byte aus dem Empfangsregister der ACIA gelesen und nach Abfrage des Sendestatus (Zeile 31), sprich nachdem das Senderegister frei ist, nach MIDI-Out geschickt (Zeile 33)

Danach springt unser Midi-Thru-Maker die IKBDSYS-Routine an und verarbeitet eventuell noch anliegende Tastaturereignisse.

In Zeile 38 (Label 'ende') wird noch nach einer ausstehenden Unterbrechung gefragt. Diese wird gegebenenfalls durch Ansprung von Zeile 25 abgearbeitet.

Liegt nichts mehr an, wird der alte Status wiederhergestellt und der Interrupt gelöscht.

Wie der aufmerksame Leser sicher schon bemerkt hat, werden die MIDI-Daten quasi verschluckt, d.h. sie werden nicht in den MIDI-BUFFER geschrieben. Aber dieses kleine Hilfsprogramm dient ja auch nicht als Ersatz für ein Sequencerprogramm.

Zur Erinnerung seien noch die MIDI-ACIA-Register genannt:

$FF FC04 .... STEUERREGISTER
$FF FC05 .... STATUSREGISTER
$FF FC06 .... SENDEREGISTER
$FF FC07 .... EMPFANGSREGISTER

Noch ein Wort zur Adresse der IKBDSYS-Interruptroutine. Es ist zwar nicht sehr fein, Betriebssystemroutinen direkt anzuspringen, aber man kommt nicht umhin, will man das Programm so kurz wie irgend möglich halten. Also die Adressen lauten für:

ROM-Version
vom 6.2.1986 ....$FC 280A
Blitter TOS ....$FC 2A0C

Viel Spaß beim Tippen und gleichzeitigem Spielen auf dem Masterkeyboard.



```
 1:  start      move.l 4(sp),a0    ; Proggröße berechnen
 2:             move.l #$100,d7    ; Basepage
 3:             add.l 12(a0),d7    ; + Textsegment
 4:             add.l 20(a0),d7    ; + Datensegment
 5:             add.l 28(a0),d7    ; + Bss
 6:             pea superinit      ; Adr. von superinit
                                     auf Stack
 7:             move #38,-(sp)     ; im Supervisor-Modus
                                     ausführen
 8:             trap #14           ; zum Bios
 9:             addq.l #6,sp       ; Stack korrigieren
10:             clr -(sp)          ; exit Code 0 auf Stack
11:             move.l d7,-(sp)    ; Programmlänge a.Stack
12:             move #49,-(sp)     ; Programmende mit
                                     Reservieren des
13:             trap #1            ; Speichers (KEEPTERM)
14:
15:  superinit move.l #mkb,$118    ; Vektor neue
                                     INT-Routine
16:             or.b #64,$fffa09   ; Empfänger Int
                                     erlauben IERB
17:             or.b #64,$fffa15   ; IMRB
18:             rts                ; fertig
19:
20:
21:  mkb        movem.l d0-d3/a0-a3/a5,-(sp) ; Register
                                               retten
22:             move.w sr,-(sp)    ; Status retten
23:             ori.w #$700,sr     ; nicht stören lassen
24:
25:  anfang     move.l #$fffc04,a0 ; MIDI-ACIA-Adresse
26:             move.b 1(a0),d3    ; hole MIDI-ACIA Status
27:             btst.b #0,d3       ; Int. von MIDI-ACIA?
28:             beq no_midi        ; nein, dann Ende
29:             move.b 2(a0),d0    ; hole Byte aus
                                     Empfängerregister
30:  wait       move.b 1(a0),d3    ; hole ACIA-Status
31:             btst.b #1,d3       ; Senderegister bereit?
32:             beq wait           ; nein, dann warte
33:             move.b d0,3(a0)    ; ja, dann sende Byte
34:
35:  no_midi    lea $0,a5          ; a5 löschen
36:             jsr $fc2a0c        ; zur IKBSYS-Routine
37:
38:  ende       btst #4,$fffffa01  ; Noch'n Interrupt?
39:             beq anfang         ; ja, dann zum Anfang
40:             move.w (sp)+,sr    ; alten Status
                                     herstellen
41:             bclr #6,$fffffa11  ; Int. löschen
42:             movem.l (sp)+,d0-d3/a0-a3/a5 ; Register
                                               herstellen
43:             rte                ; fertig
```

EINGABE VON TERMEN

Immo Wedekind

Das Programm ist in GFA-BASIC V2.0 geschrieben und dient dazu, in eigenen GFA-BASIC-Progammen Terme einzugeben, die der Computer dann ausrechnet. Es beinhaltet Klammerrechnung, Potenzieren, Dividieren, Multiplizieren, Addieren, Subtrahieren (nach Rangfolge geordnet). Ferner berücksichtigt es 'Punktrechnung vor Strichrechnung'.Im momentanen Zustand kann man bis zu 100 Klammern verwenden, was man aber in der Dimensionierung am Anfang in 'A$()' ändern kann.

Zahlen können auch bis zu 100 eingegeben werden (wie oben änderbar in 'A()'. Die Zahl 'Pi' ist im Programm festgelegt. Statt einer Zahl schreibe man in Kleinbuchstaben 'pi'. Darüber hinaus kann man auch Variablen in einen Term einfügen. Variablen sind entweder ein Buchstabe oder Zahlen bis zu 200. Der Variableninhalt muß in 'z (Buchstabe,Zahl)' stehen. Als 'Buchstabe' gelten alle ASCII-Werte bis 64. Groß- und Kleinschreibung wird beachtet!

Beispiel:

Variable A1:
Inhalt steht in 'z(1,1)'.
Variable a1:
Inhalt steht in 'z(33,1).

(Kann man gut für eine selbstgeschriebene Tabellenkalkulation nehmen!)

Wenn die Definierung zu niedrig ist: Beim Dim-Befehl 'z(255,neue Definierung), außerdem im Unterprogramm 'Mal'

Eintrag:

if p>255 or o>200

ersetzen durch:

if p>255 or o>neue Definierung.

Noch einmal zu den Klammern! Falls eine schließende oder öffnende Klammer fehlt, gibt das Programm eine Fehlermeldung aus. Dasselbe tut es bei zu hoher Variablendefinierung.

```
 0:  (c) MAXON Computer GmbH, Eschborn
 1:  Cls
 2:  Input A$ ' Eingabe des Terms
 3:  F$=A$
 4:  Dim D%(Len(A$)),A(100),A$(100),Z(255,200)
 5:  Z(1,1)=12
 6:  D%(0)=-1
 7:  Do
 8:    Add J,1
 9:    B$=Mid$(F$,J,1)
10:    If B$="("
11:      Add Kl,1
12:      A$(Kl)=""
13:    Else
14:      If B$<>")"
15:        A$(Kl)=A$(Kl)+B$
16:      Endif
17:    Endif
18:    If B$=")"
19:      A$=A$(Kl)
20:      @Mal
21:      Sub Kl,1
22:      A$(Kl)=A$(Kl)+Str$(Er)
23:    Endif
24:    If J=Len(F$) And (Kl>0 Or Kl<0)
25:      Print "Fehler bei Eingabe!"
26:      End
27:    Endif
28:    Exit If J=Len(F$) And Kl=0
29:  Loop
30:  A$=A$(0)
31:  @Mal
32:  Print Er
33:  Procedure Mal
34:    Local C$,B$,I,Ii
35:    Do
36:      Add I,1
37:      B$=Mid$(A$,I,2)
38:      B=Asc(Left$(B$,1))
39:      If B=42 Or B=43 Or B=45 Or B=47 Or B=94
40:        If Right$(B$,1)="-"
41:          C$=C$+"'"
```

→

```
 42:          Mid$(A$,I+1,1)="'"
 43:        Endif
 44:      Endif
 45:      Exit If I=Len(A$)
 46:    Loop
 47:    I=0
 48:    Do
 49:      Add I,1
 50:      B$=Mid$(A$,I,1)
 51:      If B$=Chr$(94)
 52:        C$=C$+"^"
 53:      Endif
 54:      Exit If I=Len(A$)
 55:    Loop
 56:    I=0
 57:    Do
 58:      Add I,1
 59:      B$=Mid$(A$,I,1)
 60:      If B$=Chr$(42)
 61:        C$=C$+"*"
 62:      Endif
 63:      If B$=Chr$(47)
 64:        C$=C$+"/"
 65:      Endif
 66:      Exit If I=Len(A$)
 67:    Loop
 68:    I=0
 69:    Do
 70:      Add I,1
 71:      B$=Mid$(A$,I,1)
 72:      If B$=Chr$(43)
 73:        C$=C$+"+"
 74:      Endif
 75:      If B$=Chr$(45)
 76:        C$=C$+"-"
 77:      Endif
 78:      Exit If I=Len(A$)
 79:    Loop
 80:    I=0
 81:    Ii=0
 82:    Do
 83:      Add I,1
 84:      B=Asc(Mid$(A$,I,1))
 85:      If B=42 Or B=43 Or B=47 Or B=45 Or B=94
              Or B=39
 86:        Add Ii,1
 87:        D%(Ii)=I
 88:      Endif
 89:      Exit If I=Len(A$)
 90:    Loop
 91:    Add Ii,1
 92:    D%(Ii)=I
 93:    I=0
 94:    Do
 95:      Add I,1
 96:      If Asc(Mid$(A$,D%(I-1)+1,D%(I)-1))-64<0
 97:        A(I)=Val(Mid$(A$,D%(I-1)+1,D%(I)-1))
 98:      Else
 99:        If Mid$(A$,D%(I-1)+1,D%(I)-1)="pi"
100:          A(I)=Pi
101:          Goto R
102:        Endif
103:        P=Asc(Mid$(A$,D%(I-1)+1,1))-64
104:        O=Val(Mid$(A$,D%(I-1)+2,D%(I)-1))
105:        If P>255 Or O>200
106:          Print "Eine Variable hat eine zu
                     groPe Adresse!"
107:          End
108:        Endif
109:        A(I)=Z(P,O)
110:        R:
111:      Endif
112:      Exit If I=Len(C$)+1
113:    Loop
114:    I=0
115:    C$=C$+" "
116:    Ii=1
117:    Do
118:      Add I,1
119:      If Mid$(C$,Ii,1)=Mid$(A$,D%(I),1)
120:        B=Asc(Mid$(C$,Ii,1))
121:        H=I+1
122:        If A(I+1)<0
123:          G=1
124:        Endif
125:        If B=39
126:          A(I)=A(I)-A(I+1)
127:          A(I+1)=A(I)
128:        Endif
129:        If B=94
130:          A(I)=A(I)^A(I+1)
131:          A(H)=A(I)
132:        Endif
133:        If B=42
134:          A(I)=A(I)*A(I+1)
135:          A(H)=A(I)
136:        Endif
137:        If B=47
138:          A(I)=A(I)/A(I+1)
139:          A(H)=A(I)
140:        Endif
141:        If B=43
142:          A(I)=A(I)+A(I+1)
143:          A(H)=A(I)
144:        Endif
145:        If B=45
146:          A(I)=A(I)-A(I+1)
147:          A(H)=A(I)
148:        Endif
149:        If G=1
150:          A(I+2)=A(I)
151:          G=0
152:        Endif
153:        Er=A(I)
154:        Add Ii,1
155:      Endif
156:      If I=Len(C$)+1
157:        I=0
158:      Endif
159:      ' Ende wenn 'Ii' = länge von 'C$' ist
160:      Exit If Ii=Len(C$)
161:    Loop
162:    ' Zurück
163: Return
```

ZÄHLEN VON WÖRTERN

Otto Glöggler

FOLGENDES PROGRAMM BASIERT AUF EINEM ARTIKEL VON JON BENTLEY [1], DER DARIN EIN BEISPIEL VON DONALD E. KNUTH FÜR DEN VON IHM ENTWICKELTEN PROGRAMMIERSTIL WEB DARSTELLT.

Als Problem stellt sich dabei die Aufgabe, in einer Textdatei die k häufigsten Wörter festzustellen und der Häufigkeit nach alphabetisch sortiert auszugeben. Ein derartiges Werkzeug läßt sich bei schriftstellerischer Betätigung durchaus sinnvoll anwenden, da man dadurch zu viele Wiederholungen vermeiden kann. Der eigentlich interessante Aspekt der vorgestellten Lösung ist allerdings die Benutzung einer speziellen Datenstruktur, die sich auch für ähnliche Anwendungen verwenden läßt.

Bevor man an die Lösung eines Problems herangeht, ist eine möglichst vollständige Spezifikation erforderlich. Ein etwas genaueres Betrachten des obigen Problems läßt noch einige Fragen offen wie z.B.: Was ist ein Wort? Was passiert, wenn der Text weniger als k Worte enthält? Wie ist zu verfahren, wenn mehr als k Worte mit gleicher Häufigkeit auftreten?

Beginnen wir mit der Definition eines Wortes. Ein Wort bedeutet im folgenden eine zusammenhängende Folge von Buchstaben, wobei nicht zwischen Groß- und Kleinbuchstaben unterschieden wird. Die Folge muß maximal sein, d.h. sie kann nicht verlängert werden, ohne ein Zeichen ungleich einem Buchstaben aufzunehmen. Was unter einem Buchstaben zu verstehen ist, kann selbstverständlich ebenfalls eigens definiert werden. In der Regel wird er wohl der üblichen Interpretation entsprechen, d.h. eines der Zeichen 'A' .. 'Z'. Bei unterschiedlichen nationalen Alphabeten lassen sich aber zusätzliche Zeichen aufnehmen, so wie im deutschen Alphabet die Umlaute.

Für die restlichen offenen Fragen legen wir fest: Sind mehr als *k* Wörter mit gleicher Häufigkeit vorhanden, so wird nach Ausgabe von *k* Wörtern abgebrochen. Enthält der Text weniger als *k* Worte, werden eben nur die enthaltenen berücksichtigt. Die Anzahl der auszugebenden Worte wird im Dialog abgefragt.

Lösung

Beim Lesen des Textes wissen wir natürlich nicht, welche Wörter auftauchen können. Wir müssen daher ein Wörterbuch anlegen, das jedes Wort aufnimmt und die Häufigkeit zählt. Das Sortieren nach der Häufigkeit wird erst vor der Ausgabe erledigt, da der Aufwand während des Zählvorgangs zu groß ist.

Die Hauptstruktur des Programms stellt sich damit folgendermaßen dar:

- Bestimmen von *k* und Zuweisen der Dateien.
- Initialisieren von Variablen.
- Lesen der Eingabedatei und Zählen der Wörter.
- Sortieren nach der Häufigkeit.
- Ausgabe des Ergebnisses.

Für das Lesen der Wörter wird die Prozedur *LeseWort* benutzt. Sie liest immer ein komplettes Wort, wobei für ein Wort eine maximale Länge angenommen wird. Das Wort wird in einen Puffer abgelegt und längere Wörter einfach abgeschnitten. Das Array *ZeichenCode* ordnet einem beliebigen Zeichen den Wert 0 zu, wenn es nicht im Alphabet vorkommt; falls es das k-te Zeichen im Alphabet ist, gilt:

```
ZeichenCode (Ord (Zeichen)) = k
```

Wurde ein Wort gelesen, müssen wir in unserem Wörterbuch nachschauen, ob es bereits vorhanden ist und es ggfs. aufnehmen. Das Wörterbuch soll uns dabei eine schnelle Suche erlauben, in der Lage sein, Wörter variabler Länge aufzunehmen, und eine alphabetische Sortierung erleichtern.

Die schnelle Suche legt auf jeden Fall ein Hash-Verfahren nahe, wegen der variablen Wort länge können aber keine Tabelleneinträge fester Länge verwendet werden, ohne übermäßig viel Platz zu verschwenden.

Die meisten Sprachen haben gerade die Eigenschaft, daß die häufig benutzen Wörter kurz sind. Es wird daher eine Va-

```
Häufigkeitsverteilung von Wörtern
Wieviel Wörter sollen ausgegeben werden ?
Standardwert durch 0
10
Eingabedatei : Enter 'in' filename > woerter.txt

Ausgabedatei : Enter 'out' filename > worttest
```

Eingabe der benötigten Parameter.....

riante einer Datenstruktur benutzt, die von Frank M. Liang [2] stammt und *Hash-Trie* genannt wird. Das Suchen von Einträgen geht relativ schnell, das Aufnehmen hingegen braucht etwas länger.

Der Hash-Trie

Ein *Trie* repräsentiert eine Menge von Wörtern und alle Präfixe dieser Wörter. Zur Vereinfachung wollen wir auch alle nicht leeren Präfixe von Wörtern in unserem Wörterbuch als Wörter bezeichnen, obwohl sie vielleicht gar nicht als solche in der Eingabedatei vorkommen. Jedes Wort wird dann durch einen Zeiger dargestellt, der ein Index für die vier Arrays *Naechstes, Geschwister, Anzahl* und *Zeichen* ist. Im weiteren werde ich für diese Arrays die Abkürzungen *N, G, A* und *Z* benutzen.

Wörter, die nur aus einem Buchstaben bestehen, werden durch die Zeigerwerte *1* bis *Alphabetgroesse (AG)* dargestellt. Längere Wörter sind rekursiv definiert:

Wenn der Wert von *p* das Wort *w* darstellt und wenn *1 <= c <= AG* ist, dann wird das Wort, das durch Zusammenhängen von *w* und dem *c-ten* Buchstaben des Alphabetes entsteht, durch den Wert *N(p)+c* dargestellt.

Versuchen wir einfach, das Wort "atari" darzustellen. Nehmen wir zunächst an, daß folgende Einträge vorliegen:

N(1) =1000, N(1020) = 2000, N(2001) = 3000 und N(3018) =4000.

Dann gilt für 'atari'

a = 1 at = N(1) + 20 = 1000 + 1 = 1020 ata = N(1020) + 1 = 2000 + 1 = 2001 atar = N(2001) + 18 = 3000 + 18 = 3018 atari = N(3018) + 9 = 4000 + 9 = 4009

Gibt es kein längeres Wort, das mit 'atari' beginnt, ist der Eintrag in *N(4009)= 0.*

Die Struktur beinhaltet zusätzlich redundante Information, um das Ändern und Durchlaufen zu erleichtern.

Wenn *N(p) <> 0* ist, dann gilt *N(N(p)) = p*.

Wenn weiterhin *q = N(p) + c* ein "Kind" von *p* ist, dann gilt *Z(q) = c*.

In unserem Beispiel gilt somit *Z(1) = 1, Z(1020) = 20*, usw.

Dies erlaubt es, vom Kind zum Vater zugehen, da *N(q-Z(q)) = N(N(p)+c-Z(q)) = N(N(p)) = p* ist.

Kinder mit gleichem Vater sind durch Geschwisterzeiger verknüpft: Das "älteste" Kind von *p* ist *G(N(p))*, und das nächst ältere ist *G(G(N(p)))*, usw. Der Geschwisterzeiger des jüngsten Kindes ist *N(p)*.

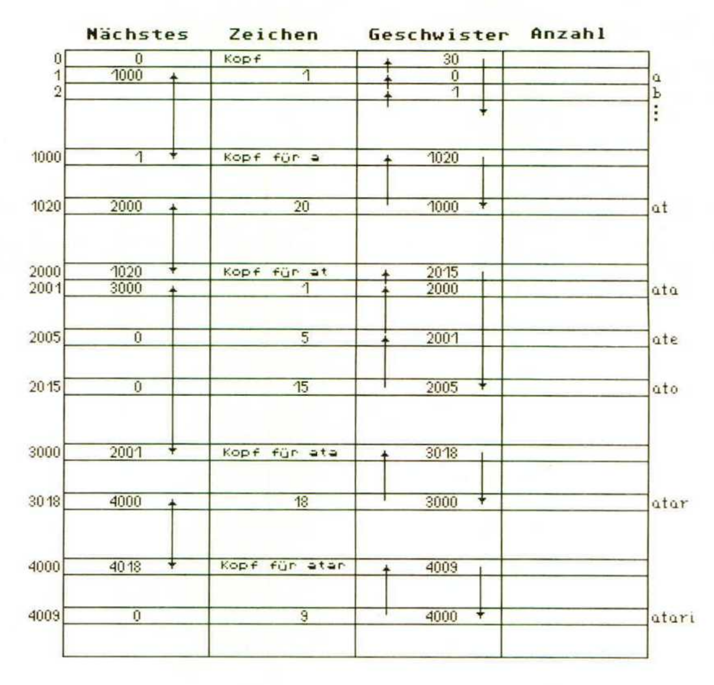
Setzen wir unser Beispiel fort, so bedeutet das für den Fall, daß alle Wörter mit Anfang 'at' entweder mit 'ata', 'ate' oder 'ato' beginnen:

G(N(1020)) = G(2000) = 2015 (= 'ato').

Zur Erinnerung: 'at' wird durch den Wert 1020 repräsentiert. *G(2015) = 2005 (= 'ate'). G(2005) = 2001 (= 'ata')* und *G(2001) = 2000.*

Wenn *N(p) <> 0* ist, dann wird der Eintrag in Position *N(p)* der "Kopf" von *p*s Kinder genannt. Der spezielle Wert *KopfEintrag* erscheint im Zeichen-Feld eines jeden Kopfeintrags.

Wenn *p* ein Wort darstellt, gibt *A(p)* die bisherige Häufigkeit des Wortes an. Die Anzahl von einem Kopfeintrag ist undefiniert. Für nicht benutzte Positionen *p* gilt *Z(p) = LeerEintrag*. In diesem Fall sind *N(p), G(p)* und *A(p)* nicht definiert.



	Nächstes	Zeichen	Geschwister	Anzahl	
0	0	Kopf	30		
1	1000	1	0		a
2			1		b
					⋮
1000	1	Kopf für a	1020		
1020	2000	20	1000		at
2000	1020	Kopf für at	2015		
2001	3000	1	2000		ata
2005	0	5	2001		ate
2015	0	15	2005		ato
3000	2001	Kopf für ata	3018		
3018	4000	18	3000		atar
4000	4018	Kopf für atar	4009		
4009	0	9	4000		atari

Programm

Da der benutzte Compiler die Größe der globalen Daten einschränkt, werden für die Array *Naechstes, Geschwister, Zeichen* und *Anzahl* eigene Datenmoduln vereinbart. Für die Ein- und Ausgabe werden nur Prozeduren des Standardmoduls *InOut* benutzt, wie sie von N. Wirth vorgeschlagen wurden und z.B. in [2] beschrieben sind. Eine Übersetzung sollte damit mit jedem Modula-2-Compiler möglich sein.

Eine der Hauptaufgaben des Einfügealgorithmus ist die Suche nach einem Platz für einen neuen Kopf. Die Theorie von Hashverfahren zeigt, daß es günstig ist, den *n-ten* Kopf nahe dem Platz *x(n) = (a*n)mod t* abzulegen, wobei *t = Feldgroesse - 2*AG* und *a* eine Primzahl ist, so daß *a/t* ungefähr *(Wurzel(5)-1)/2* ist. Für eine Feldgröße von 16381 ergibt sich damit 10091.

Da eine Hashtabelle auch überlaufen kann, setzen wir mit Toleranz eine Grenze für die Anzahl der Versuche zum Finden eines freien Platzes.

Das Sortieren nach der Häufigkeit wird gesondert durchgeführt. Beim Sortieren werden die meisten Häufigkeiten klein sein, so daß es ausreicht, einige verkettete Listen 'Sortiert' mit den Wörtern kleiner Häufigkeit und eine separaten Liste, die den Rest aufnimmt, einzurichten. Für den Aufbau der Listen werden einfach die Geschwisterzeiger benutzt.

Beim Durchlaufen des Wörterbuchs in umgekehrter alphabetischer Reihenfolge, werden die Verkettungen von Geschwister so geändert, daß die Werte von *S(f), G(S(f)), ...* auf die Wörter mit Häufigkeit *f* in alphabetischer Reihenfolge zeigen. Wenn *f = GrosseHäufigkeit* ist, d.h. die Wörter in die separate Liste kommen, müssen sie zusätzlich sortiert nach ihrer Häufigkeit verkettet werden.

Nach dem Sortieren enthalten die Listen 'Sortiert' alle Wörter der Eingabedatei in absteigender Reihenfolge. Die Wörter mit gleicher Häufigkeit sind durch das Geschwisterfeld verkettet. Die Prozedur *DrukkeVerteilung* gibt dann die ersten *k* Wörter aus.

Die 10 häufigsten Wörter:

die	43
der	37
n	31
ist	24
p	24
und	24
von	24
das	20
wörter	18
in	17

Es wurden insgesamt 1269 Wörter und 148 Zeilen gelesen.

..... Ausgabe in eine ASCIIDatei

Literatur

[1] Jon Bentley, Programming Pearls, Communications of the ACM, Vol. 29, No. 6, 1986

[2] M. Dal Cin, J. Lutz, T. Risse, Programmierung in Modula-2, B.G. Teubner, Stuttgart 1986, 2. Aufl.

```
  1:  (**********************************************)
  2:  (* Zählen von Wörtern in einem Text und Ausgabe *)
  3:  (* nach der Häufigkeit sortiert.                *)
  4:  (* (c) MAXON Computer GmbH                      *)
  5:  (**********************************************)
  6:  MODULE Woerter;
  7:
  8:  FROM InOut      IMPORT EOL, Done, OpenInput,
                             OpenOutput, CloseInput,
  9:                         CloseOutput, Read, ReadCard,
                             WriteLn, WriteString,
 10:                         Write, WriteCard;
 11:  FROM Wort0Daten IMPORT AlphabetGroesse,
                             DefaultAnzahl,
 12:                         HashTabellenGroesse,
                             MaximaleWortlaenge,
 13:                         LeerEintrag, KopfEintrag,
 14:                         Alpha, Toleranz,
 15:                         GrenzeUeberschreiten,
                             GrosseHaeufigkeit,
 16:                         BUCHSTABENCODE0,
                             BUCHSTABENCODE1,
 17:                         WORTLAENGE, ZEIGER,
 18:                         MaximaleAnzahl;
 19:
 20:  (* Da globale Daten beschränkt sind, werden eigene
        Datenmodule benutzt *)
 21:
 22:  FROM Wort1Daten IMPORT Geschwister;
 23:  FROM Wort2Daten IMPORT Anzahl;
 24:  FROM Wort3Daten IMPORT Naechstes;
 25:  FROM Wort4Daten IMPORT Zeichen;
 26:  FROM Wort5Daten IMPORT Sortiert;
 27:
 28:  VAR
 29:
 30:    AnzahlWoerter, GesamteWoerter,
        BisherigeWortanzahl, BisherigeZeilen
 31:      : CARDINAL;
 32:    Kleinbuchstabe, Grossbuchstabe
 33:      : ARRAY [1 .. AlphabetGroesse] OF CHAR;
 34:    ZeichenCode
 35:      : ARRAY [0 .. 255] OF BUCHSTABENCODE0;
 36:    Wortpuffer
 37:      : ARRAY [1 .. MaximaleWortlaenge] OF
                                  BUCHSTABENCODE1;
 38:    Wortlaenge
 39:      : WORTLAENGE;
 40:    WortGekuerzt, WortUebergangen
 41:      : BOOLEAN;
 42:    ToleranzUeberschritten
 43:      : CARDINAL;
 44:    KopfPlatz, Wort
 45:      : ZEIGER;
 46:
 47:    (*********************************)
 48:    (**) PROCEDURE Dateizuweisung;(**)
 49:    (*********************************)
 50:    (*  erledigt die Zuweisung der   *)
        (*     Ein-und Ausgabedatei      *)
 51:
 52:    BEGIN
 53:      WriteString ('Eingabedatei : ');
 54:      OpenInput ("TXT");
 55:      WriteLn;
 56:      WriteString ('Ausgabedatei : ');
 57:      OpenOutput ("LST");
 58:    END Dateizuweisung;
 59:
 60:    (*********************************)
 61:    (**) PROCEDURE Initialisieren; (**)
 62:    (*********************************)
 63:    (*  initialisiert die Tabellen    *)
 64:
 65:    VAR
 66:      I : CARDINAL;
 67:
 68:    BEGIN
 69:      Kleinbuchstabe [ 1] := 'a';
          Grossbuchstabe [ 1] := 'A';
 70:      Kleinbuchstabe [ 2] := 'b';
          Grossbuchstabe [ 2] := 'B';
 71:      Kleinbuchstabe [ 3] := 'c';
          Grossbuchstabe [ 3] := 'C';
 72:      Kleinbuchstabe [ 4] := 'd';
          Grossbuchstabe [ 4] := 'D';
```

```
 73:      Kleinbuchstabe [ 5] := 'e';
          Grossbuchstabe [ 5] := 'E';
 74:      Kleinbuchstabe [ 6] := 'f';
          Grossbuchstabe [ 6] := 'F';
 75:      Kleinbuchstabe [ 7] := 'g';
          Grossbuchstabe [ 7] := 'G';
 76:      Kleinbuchstabe [ 8] := 'h';
          Grossbuchstabe [ 8] := 'H';
 77:      Kleinbuchstabe [ 9] := 'i';
          Grossbuchstabe [ 9] := 'I';
 78:      Kleinbuchstabe [10] := 'j';
          Grossbuchstabe [10] := 'J';
 79:      Kleinbuchstabe [11] := 'k';
          Grossbuchstabe [11] := 'K';
 80:      Kleinbuchstabe [12] := 'l';
          Grossbuchstabe [12] := 'L';
 81:      Kleinbuchstabe [13] := 'm';
          Grossbuchstabe [13] := 'M';
 82:      Kleinbuchstabe [14] := 'n';
          Grossbuchstabe [14] := 'N';
 83:      Kleinbuchstabe [15] := 'o';
          Grossbuchstabe [15] := 'O';
 84:      Kleinbuchstabe [16] := 'p';
          Grossbuchstabe [16] := 'P';
 85:      Kleinbuchstabe [17] := 'q';
          Grossbuchstabe [17] := 'Q';
 86:      Kleinbuchstabe [18] := 'r';
          Grossbuchstabe [18] := 'R';
 87:      Kleinbuchstabe [19] := 's';
          Grossbuchstabe [19] := 'S';
 88:      Kleinbuchstabe [20] := 't';
          Grossbuchstabe [20] := 'T';
 89:      Kleinbuchstabe [21] := 'u';
          Grossbuchstabe [21] := 'U';
 90:      Kleinbuchstabe [22] := 'v';
          Grossbuchstabe [22] := 'V';
 91:      Kleinbuchstabe [23] := 'w';
          Grossbuchstabe [23] := 'W';
 92:      Kleinbuchstabe [24] := 'x';
          Grossbuchstabe [24] := 'X';
 93:      Kleinbuchstabe [25] := 'y';
          Grossbuchstabe [25] := 'Y';
 94:      Kleinbuchstabe [26] := 'z';
          Grossbuchstabe [26] := 'Z';
 95:      Kleinbuchstabe [27] := 'ä';
          Grossbuchstabe [27] := 'Ä';
 96:      Kleinbuchstabe [28] := 'ö';
          Grossbuchstabe [28] := 'Ö';
 97:      Kleinbuchstabe [29] := 'ü';
          Grossbuchstabe [29] := 'Ü';
 98:      Kleinbuchstabe [30] := 'P';
          Grossbuchstabe [30] := 'P';
 99:
100:      FOR I := 0 TO 255 DO
101:        ZeichenCode [I] := 0;
102:      END;
103:      FOR I := 1 TO AlphabetGroesse DO
104:        ZeichenCode [ORD (Kleinbuchstabe [I])]:= I;
105:        ZeichenCode [ORD (Grossbuchstabe [I])]:= I;
106:      END;
107:      WortGekuerzt := FALSE;
108:
109:      FOR I := AlphabetGroesse+1 TO
                          HashTabellenGroesse DO
110:        Zeichen [I] := LeerEintrag;
111:      END;
112:      FOR I := 1 TO AlphabetGroesse DO
113:        Zeichen [I]     := I;
114:        Naechstes [I]   := 0;
115:        Anzahl [I]      := 0;
116:        Geschwister [I] := I - 1;
117:      END;
118:
119:      Zeichen [0]     := KopfEintrag;
120:      Naechstes [0]   := 0;
121:      Geschwister [0] := AlphabetGroesse;
122:      KopfPlatz       := 0;
123:      WortUebergangen := FALSE;
124:    END Initialisieren;
125:
126:    (*********************************)
127:    (**) PROCEDURE LeseWortanzahl (**)
        (**)   (VAR A : CARDINAL);    (**)
128:    (*********************************)
129:    BEGIN
130:      WriteString ('Wieviel Wörter sollen
                       ausgegeben werden ? ');
```

→

```
131:      WriteLn;
132:      WriteString ('Standardwert durch 0');
133:      WriteLn;
134:      ReadCard (A);
135:      WriteLn;
136:      IF A = 0 THEN
137:        A := DefaultAnzahl;
138:      END;
139:    END LeseWortanzahl;
140:
141:
142:    (**************************)
143:    (**) PROCEDURE LeseWort;(**)
144:    (**************************)
145:
146:    VAR
147:      Zeichen
148:        : CHAR;
149:
150:    BEGIN
151:      Wortlaenge := 0;
152:      Read (Zeichen);
153:      IF Done THEN
154:        WHILE ZeichenCode [ORD (Zeichen)] = 0 DO
155:          Read (Zeichen);
156:          IF Zeichen = EOL THEN
157:            INC (BisherigeZeilen);
158:          END;
159:          IF NOT Done THEN
160:            RETURN;
161:          END;
162:        END;
163:        REPEAT
164:          IF Wortlaenge = MaximaleWortlaenge THEN
165:            WortGekuerzt := TRUE;
166:          ELSE
167:            INC (Wortlaenge);
168:            Wortpuffer [Wortlaenge] := ZeichenCode
                                          [ORD (Zeichen)];
169:          END;
170:          Read (Zeichen);
171:        UNTIL (ZeichenCode [ORD (Zeichen)] = 0) OR
                                                 (NOT Done);
172:        IF Zeichen = EOL THEN
173:          INC (BisherigeZeilen);
174:        END;
175:      END;
176:    END LeseWort;
177:
178:    (********************************************)
179:    (**) PROCEDURE DruckeWort (Wort : ZEIGER);(**)
180:    (********************************************)
181:
182:    VAR
183:      Vorgaenger
184:        : ZEIGER;
185:      I
186:        : [1 .. MaximaleWortlaenge];
187:
188:    BEGIN
189:      (* Wort zeigt auf das Wort, d.h. den letzten
            Buchstaben *)
190:      Wortlaenge := 0;
191:      Vorgaenger := Wort;
192:      Write (' ');
193:      REPEAT
194:        INC (Wortlaenge);
195:        Wortpuffer [Wortlaenge] := Zeichen
                                      [Vorgaenger];
196:        (* weiterschalten zum Prefix *)
197:        Vorgaenger := Naechstes [Vorgaenger -
                              Zeichen [Vorgaenger]];
198:      UNTIL Vorgaenger = 0;
199:
200:      FOR I := Wortlaenge TO 1 BY -1 DO
201:        Write (Kleinbuchstabe [Wortpuffer [I]]);
202:      END;
203:
204:      IF Anzahl [Wort] < MaximaleAnzahl THEN
205:        WriteString ('  ');
206:        WriteCard (Anzahl [Wort], 1)
207:      ELSE
208:        WriteString ('  ');
209:        WriteCard (MaximaleAnzahl, 1);
210:        WriteString (' oder gröPer');
211:      END;
212:      WriteLn;
213:    END DruckeWort;
214:
215:    (***********************************)
216:    (**) PROCEDURE DruckeVerteilung (**)
        (**)    (Anzahl : CARDINAL);    (**)
217:    (***********************************)
218:
219:    VAR
220:      Haeufigkeit
221:        : [1 .. GrosseHaeufigkeit + 1];
222:      Wort
223:        : ZEIGER;
224:
225:    BEGIN
226:      Haeufigkeit := GrosseHaeufigkeit + 1;
227:      Wort        := LeerEintrag;
228:      REPEAT
229:        WHILE Wort = LeerEintrag DO
230:          IF Haeufigkeit = 1 THEN
231:            RETURN;
232:          END;
233:          DEC (Haeufigkeit);
234:          Wort := Sortiert [Haeufigkeit];
235:        END;
236:        DruckeWort (Wort);
237:        DEC (Anzahl);
238:        Wort := Geschwister [Wort];
239:      UNTIL Anzahl = 0;
240:    END DruckeVerteilung;
241:
242:    (*******************************************)
243:    (**) PROCEDURE SucheWort () : ZEIGER;    (**)
244:    (*******************************************)
245:    (* Sucht ein Wort; falls vorhanden wird der*)
        (*        Tabellenindex geliefert,         *)
246:    (*   wenn noch nicht vorhanden, wird es    *)
        (* aufgenommen. Ist kein Platz mehr frei,  *)
247:    (*  wird als Zeigerwert 0 zurückgeliefert  *)
248:
249:    VAR
250:      I
251:        : [1 .. MaximaleWortlaenge];
252:      AktWortPos, NexWortPos, NeuerPlatz,
          ZuVerschieben
253:        : ZEIGER;
254:      AktZeichen
255:        : BUCHSTABENCODE1;
256:      LetzterNeuerPlatz, Delta
257:        : CARDINAL;
258:      EintragGefunden
259:        : BOOLEAN;
260:
261:    BEGIN
262:      I := 1;
263:      AktWortPos := Wortpuffer [I];
264:      WHILE I < Wortlaenge DO
265:        INC (I);
266:
267:        (* weiterschalten zum Kind *)
268:
269:        AktZeichen := Wortpuffer [I];
270:        IF Naechstes [AktWortPos]=LeerEintrag THEN
271:
272:          (* bisher noch kein Kind vorhanden,
                deshalb aufnehmen, falls *)
273:          (* genügend Platz vorhanden *)
274:
275:          (* zumächst weiterschalten um alpha *)
276:
277:          IF KopfPlatz < HashTabellenGroesse -
                2*AlphabetGroesse - Alpha THEN
278:            KopfPlatz := KopfPlatz + Alpha
279:          ELSE
280:            KopfPlatz := KopfPlatz + Alpha -
                HashTabellenGroesse + 2*AlphabetGroesse;
281:          END;
282:
283:          (* jetzt gilt: *)
284:          (* AlphabetgröPe < KopfPlatz <=
                TabellengröPe - AlphabetgröPe  *)
285:          (* dann wird bestimmt, wieviele Versuche
                erlaubt sind, um einen  *)
286:          (* freien Platz zu finden, indem als
                obere Grenze um errechneten *)
287:          (* neuen Platz die Toleranzschwelle
                addiert wird.  *)
```

→

```
288:
289:        NeuerPlatz := KopfPlatz + KopfEintrag;
290:        IF NeuerPlatz <= HashTabellenGroesse -
              AlphabetGroesse - Toleranz THEN
291:          LetzterNeuerPlatz := NeuerPlatz +
                                  Toleranz
292:        ELSE
293:          LetzterNeuerPlatz := NeuerPlatz +
                           Toleranz -
294:                       HashTabellenGroesse +
                           2*AlphabetGroesse;
295:        END;
296:
297:        (* prüfen, ob neuer Platz frei; *)
298:        (* wenn nicht, einfach nächstes
              Feld versuchen *)
299:
300:        REPEAT
301:          IF CARDINAL (NeuerPlatz) =
                LetzterNeuerPlatz THEN
302:            (* kein freier Platz gefunden *)
303:            RETURN 0;
304:          END;
305:          IF NeuerPlatz = HashTabellenGroesse -
                AlphabetGroesse THEN
306:            NeuerPlatz := KopfEintrag;
307:          ELSE
308:            INC (NeuerPlatz);
309:          END;
310:        UNTIL (Zeichen [NeuerPlatz]=LeerEintrag)
                 AND
311:             (Zeichen [NeuerPlatz+AktZeichen] =
                  LeerEintrag);
312:
313:        (* Eintragen der Kettungsverweise *)
314:
315:        Naechstes [AktWortPos]   := NeuerPlatz;
316:        Naechstes [NeuerPlatz]   := AktWortPos;
317:        AktWortPos               := NeuerPlatz +
                                        AktZeichen;
318:        Zeichen [NeuerPlatz]     := KopfEintrag;
319:        Zeichen [AktWortPos]     := AktZeichen;
320:        Geschwister [NeuerPlatz] := AktWortPos;
321:        Geschwister [AktWortPos] := NeuerPlatz;
322:        Anzahl [AktWortPos]      := 0;
323:        Naechstes [AktWortPos]   := LeerEintrag;
324:
325:      ELSE
326:
327:        (* bereits ein Kind vorhanden *)
328:
329:        NexWortPos := Naechstes [AktWortPos] +
                         AktZeichen;
330:        IF Zeichen [NexWortPos] # AktZeichen THEN
331:
332:          (* nicht das richtige Kind *)
333:
334:          IF Zeichen [NexWortPos] # LeerEintrag
               THEN
335:
336:            (* die Familie von Wort an einen
                 Platz verschieben, wo alle *)
337:            (* Kinder Platz haben *)
338:
339:            EintragGefunden := FALSE;
340:
341:            (* Neuen Platz berechnen *)
342:
343:            IF KopfPlatz < HashTabellenGroesse -
                 2*AlphabetGroesse - Alpha THEN
344:              KopfPlatz := KopfPlatz + Alpha;
345:            ELSE
346:              KopfPlatz := KopfPlatz + Alpha -
                             HashTabellenGroesse +
                             2*AlphabetGroesse;
347:            END;
348:            NeuerPlatz := KopfPlatz+KopfEintrag;
349:            IF NeuerPlatz <= HashTabellenGroesse
                 - AlphabetGroesse - Toleranz THEN
350:              LetzterNeuerPlatz := NeuerPlatz
                                     + Toleranz;
351:            ELSE
352:              LetzterNeuerPlatz := NeuerPlatz
                                  + Toleranz
353:                              - HashTabellenGroesse
                                  + 2*AlphabetGroesse;
354:            END;
355:
356:            REPEAT
357:              IF CARDINAL (NeuerPlatz) =
                   LetzterNeuerPlatz THEN
358:                INC (ToleranzUeberschritten);
359:                RETURN 0;
360:              END;
361:              IF NeuerPlatz = HashTabellenGroesse
                   - AlphabetGroesse THEN
362:                NeuerPlatz := KopfEintrag;
363:              ELSE
364:                INC (NeuerPlatz);
365:              END;
366:
367:              (* prüfen, ob auch alle Geschwister
                   Platz haben *)
368:
369:              IF Zeichen [NeuerPlatz + AktZeichen]
                   = LeerEintrag THEN
370:                ZuVerschieben := Naechstes
                                  [AktWortPos];
371:                Delta := NeuerPlatz -
                          ZuVerschieben;
372:                WHILE (Zeichen [CARDINAL
                       (ZuVerschieben) + Delta] =
                       LeerEintrag) AND
373:                   (Geschwister [ZuVerschieben]
                       # Naechstes [AktWortPos])
                         DO
374:                  ZuVerschieben := Geschwister
                                  [ZuVerschieben];
375:                END;
376:                IF Zeichen [ZuVerschieben + Delta]
                     = LeerEintrag THEN
377:                  EintragGefunden := TRUE;
378:                END;
379:              END;
380:
381:            UNTIL EintragGefunden;
382:
383:           (* die komplette Familie um delta
               verschieben *)
384:
385:            NexWortPos      := NeuerPlatz +
                                  AktZeichen;
386:            ZuVerschieben := Naechstes
                                  [AktWortPos];
387:            Delta          := NeuerPlatz -
                                  ZuVerschieben;
388:            REPEAT
389:              Geschwister [ZuVerschieben + Delta]
                   := Geschwister [ZuVerschieben]
                   + Delta;
390:              Zeichen [ZuVerschieben + Delta]
                   := Zeichen [ZuVerschieben];
391:              Zeichen [ZuVerschieben]
                   := LeerEintrag;
392:              Anzahl [ZuVerschieben + Delta]
                   := Anzahl [ZuVerschieben];
393:              Naechstes [ZuVerschieben + Delta]
                   := Naechstes [ZuVerschieben];
394:              IF Naechstes [ZuVerschieben] #
                   LeerEintrag THEN
395:                Naechstes [Naechstes
                    [ZuVerschieben]] := ZuVerschieben
                    + Delta;
396:              END;
397:              ZuVerschieben := Geschwister
                                 [ZuVerschieben];
398:            UNTIL Zeichen [ZuVerschieben] =
                              LeerEintrag;
399:          END;
400:
401:          (* das neue Kind in die Familie
              aufnehmen, dabei die richtige *)
402:          (* Reihenfolge erhalten *)
403:
404:          NeuerPlatz := Naechstes [AktWortPos];
405:          WHILE Geschwister [NeuerPlatz] >
                                   NexWortPos DO
406:            NeuerPlatz:= Geschwister [NeuerPlatz];
407:          END;
408:          Geschwister [NexWortPos]:= Geschwister
                                       [NeuerPlatz];
409:          Geschwister [NeuerPlatz]:= NexWortPos;
410:          Zeichen [NexWortPos]    := AktZeichen;
```

→

```
411:            Anzahl [NexWortPos]       := 0;
412:            Naechstes [NexWortPos]    := LeerEintrag;
413:          END;
414:          AktWortPos := NexWortPos;
415:        END;
416:      END;
417:      RETURN AktWortPos;
418:    END SucheWort;
419:
420:
421:    (****************************************)
422:    (**)   PROCEDURE SortiereTabelle;    (**)
423:    (****************************************)
424:    (*  nach Beenden zeigt Sortiert [F]    *)
        (*  auf das 1.Wort mit Häufigkeit F,   *)
425:    (* die nächsten stehen bei Geschwister*)
        (*       [Sortiert[F]],Geschwister     *)
426:    (*  [Geschwister [Sortiert [F]]],usw. *)
427:
428:    VAR
429:      K
430:        : [1 .. GrosseHaeufigkeit];
431:      AktPos, Q, R
432:        : ZEIGER;
433:      Haeufigkeit
434:        : [0 .. MaximaleAnzahl];
435:
436:    BEGIN
437:      GesamteWoerter := 0;
438:      FOR K := 1 TO GrosseHaeufigkeit DO
439:        Sortiert [K] := 0;
440:      END;
441:      AktPos := Geschwister [0];
442:
443:      (* zum letzten Suffix von AktPos gehen *)
444:
445:      WHILE Naechstes [AktPos] # LeerEintrag DO
446:        AktPos := Geschwister [Naechstes [AktPos]];
447:      END;
448:
449:      REPEAT
450:        Haeufigkeit := Anzahl [AktPos];
451:        Q := Geschwister [AktPos];
452:        IF Haeufigkeit # 0 THEN
453:          INC (GesamteWoerter);
454:          IF Haeufigkeit < GrosseHaeufigkeit THEN
455:
456:            (* wenn eigene Liste, dann einfach in
                   die Liste einklinken *)
457:
458:            Geschwister [AktPos]    := Sortiert
                                          [Haeufigkeit];
459:            Sortiert [Haeufigkeit] := AktPos;
460:          ELSE
461:
462:            (* sonst an die richtige Stelle der
                   speziellen Liste für *)
463:            (* die groPen Häufigkeiten einfügen *)
464:
465:            R := Sortiert [GrosseHaeufigkeit];
466:            IF Anzahl [AktPos] >= Anzahl [R] THEN
467:
468:              (* entweder an den Listenanfang, wenn
                     die gröPte Häufigkeit *)
469:
470:              Geschwister [AktPos]          := R;
471:              Sortiert [GrosseHaeufigkeit]:= AktPos;
472:            ELSE
473:
474:              (* oder zuerst die richtige
                     Einfügestelle suchen *)
475:
476:              WHILE Anzahl [AktPos] < Anzahl
                                 [Geschwister[R]] DO
477:                R := Geschwister [R];
478:              END;
479:              Geschwister [AktPos] :=
                                     Geschwister [R];
480:              Geschwister [R] := AktPos;
481:            END;
482:          END;
483:        END;
484:
485:        IF Zeichen [Q] # KopfEintrag THEN
486:
487:      (* falls noch weitere Kinder vorhanden sind *)
488:
489:          AktPos := Q;
490:
491:      (* wieder zum letzten Suffix v.AktPos gehen *)
492:
493:          WHILE Naechstes [AktPos] # LeerEintrag DO
494:            AktPos := Geschwister [Naechstes
                                        [AktPos]];
495:          END;
496:        ELSE
497:
498:          (* sonst zum Prefix gehen *)
499:
500:          AktPos := Naechstes [Q];
501:        END;
502:      UNTIL AktPos = 0;
503:    END SortiereTabelle;
504:
505:  (****************************************)
506:  BEGIN
507:
508:
509:    Write (33C);
510:    Write ('E');
511:    WriteString ('Häufigkeitsverteilung von
                     Wörtern');
512:    WriteLn;
513:
514:    Initialisieren;
515:    LeseWortanzahl (AnzahlWoerter);
516:    Dateizuweisung;
517:
518:    ToleranzUeberschritten := 0;
519:    BisherigeWortanzahl := 0;
520:    BisherigeZeilen := 0;
521:    LeseWort;
522:    LOOP
523:      IF Wortlaenge = 0 THEN
524:        EXIT;
525:      END;
526:      INC (BisherigeWortanzahl);
527:      Wort := SucheWort ();
528:      IF Wort = 0 THEN
529:        WortUebergangen := TRUE;
530:
531:        (* kein Platz mehr für das neue Wort *)
532:
533:        IF ToleranzUeberschritten >
              GrenzeUeberschreiten THEN
534:          WriteString ('Es wurden mehr als ');
535:          WriteCard (GrenzeUeberschreiten, 1);
536:          WriteString (' Wörter übergangen');
537:          WriteLn;
538:          WriteString ('Es wurden bisher ');
539:          WriteCard (BisherigeWortanzahl, 1);
540:          WriteString (' Wörter und ');
541:          WriteCard (BisherigeZeilen, 1);
542:          WriteString (' gelesen');
543:          WriteLn;
544:          EXIT;
545:        END;
546:      ELSE
547:        IF Anzahl [Wort] < MaximaleAnzahl THEN
548:          INC (Anzahl [Wort]);
549:        END;
550:      END;
551:      LeseWort;
552:    END;
553:
554:    SortiereTabelle;
555:
556:    IF GesamteWoerter = 0 THEN
557:      WriteString ('Die Eingabe enthält keine
                       Wörter');
558:      WriteLn;
559:    ELSE
560:      IF GesamteWoerter < AnzahlWoerter THEN
561:        WriteString ('Wörter der Eingabedatei
                         sortiert nach der Häufigkeit');
562:        WriteLn;
563:      ELSIF AnzahlWoerter = 1 THEN
564:        WriteString ('Das häufigste Wort und die
                         Häufigkeit: ')
565:      ELSE
566:        WriteString ('Die ');
567:        WriteCard (AnzahlWoerter, 1);
568:        WriteString (' häufigsten Wörter: ');
569:        WriteLn;
```

→

```
570:      END;
571:      DruckeVerteilung (AnzahlWoerter);
572:      IF WortGekuerzt THEN
573:        WriteString ('(mindestens ein Wort ist auf
                            die Länge von ');
574:        WriteCard (MaximaleWortlaenge, 1);
575:        WriteString(' Buchstaben gekürzt worden)');
576:        WriteLn;
577:      END;
578:      IF WortUebergangen THEN
579:        WriteString ('(Einige Wörter wurden wegen
                          Speichermangel übergangen)');
580:        WriteLn;
581:        END;
582:        WriteString ('Es wurden insgesamt ');
583:        WriteCard (BisherigeWortanzahl, 1);
584:        WriteString (' Wörter und ');
585:        WriteCard (BisherigeZeilen, 1);
586:        WriteString (' Zeilen gelesen.');
587:        WriteLn;
588:      END;
589:
590:      CloseOutput;
591:      CloseInput;
592:    END Woerter.
```

TURBO - UNFILL

Die Evolution eines Algorithmus

Sven Geier

KÜRZLICH SAH ICH BEI EINEM BEKANNTEN IN EINEM "PROFESSIONELLEN" MALPROGRAMM DAS WIRKEN EINER "UNFILL"-FUNKTION. DIESE BESEITIGTE IN EINEM RECHTECKIGEN BILDSCHIRMAUSSCHNITT ALLE GEFÜLLTEN FLÄCHEN UND LIESS NUR IHRE KONTUREN STEHEN. UNGEACHTET DESSEN, DASS DAS SEHR HÜBSCH AUSSAH, ÄRGERTE MICH DOCH DIE UNHEIMLICHE LANGSAMKEIT DER FUNKTION. UND SO SETZTE ICH MICH HIN, UM EINE VERBESSERTE VERSION DIESES ALGORITHMUS' AUSZUKNOBELN; HIER IST DAS ERGEBNIS.

Doch bevor ich zur optimierten Version komme, sollten Sie sich erstmal Listing 1 ansehen: Hier habe ich auf alle Geschwindigkeitstricks verzichtet, um das Prinzip eines Unfill-Algorithmus' deutlich werden zu lassen. Das Listing ist auf Monochrombetrieb ausgelegt; um in Farbe damit arbeiten zu können, müssen allerdings nur die Variablen *Xmax%* und *Ymax%* in den ersten zwei Zeilen angepaßt werden.

Nachdem ein Bild im DEGAS-Format ausgewählt wurde, wird es in den Bildschirmspeicher geladen. Wer das Programm in Farbe benutzen möchte, könnte an dieser Stelle noch die Farben richtig setzen - allerdings würde es auch nichts ausmachen, dies zu unterlassen, da größere Flächen in einer Farbe ja sowieso ausgehöhlt werden.

Sodann tastet das Programm Zeile für Zeile ab, wobei es in jeder Zeile jeden einzelnen Punkt daraufhin überprüft, ob sich seine Farbe von der seines Nachfolgers unterscheidet. Wenn ja, haben wir gerade eine Kante erreicht und der Punkt wird schwarz gefärbt; ansonsten befinden wir uns mitten in einer Fläche und färben den Punkt daher weiß.

Wenn wir uns nach einem Durchlauf durch dieses Programm das Ergebnis ansehen, stellen wir jedoch eine kleine Unschönheit fest: waagerechte Linien werden auf zwei Punkte reduziert, was dazu führt, daß geschlossene Formen oben und unten offen erscheinen (Bild 1). Abhilfe schafft hier nur ein nochmaliges Abtasten des ursprünglichen Bildes, aber diesmal senkrecht statt waagerecht (also spalten- statt zeilenweise). Die einzigen beiden Variablen, die im Programm Verwendung finden (außer den Schleifenzählern und *Xmax%* und *Ymax%*) sind die Strings *T$* und *U$*. In *T$* wird das ursprüngliche Bild abgespeichert, bevor man die waagerechte Abtastung darauf losläßt. Das Ergebnis dieser ersten Unfill-Routine wird in *U$* abgespeichert. Sodann benutzt man *T$*, um das ursprüngliche Bild wiederherzustellen. Nachdem die zweite Abtastung über das Bild gelaufen ist, kommt der Clou: Das erste Ergebnis wird mit Put in das zweite hineinkopiert, und zwar mit Modus 7, so daß sich beide Bilder vereinen.

Bild 1: (A) Zeigt die ursprüngliche Figur, (B) und (C) die Ergebnisse einer waagrechten bzw. einer senkrechten Abtastung. (D) schließlich zeigt die Kombination von (B) und (C).

Das so erstellte Programm funktioniert prima, hat jedoch einen eklatanten Schönheitsfehler: Es ist ätzend langsam! Bei mir brauchte es im Mittel achtzehn Minuten für ein 640*400-Vollbild (compiliert, aber ohne Blitter). Damit bewegt sich die Geschwindigkeit der Routine in der Größenordnung derjenigen, die ich eingangs erwähnte, und die mich auf die Idee brachte, eine Optimierung zu versuchen.

Die Optimierungsideen sind im wesentlichen folgende:

1) Ein Punkt wird im endgültigen Bild gesetzt, wenn sich seine Farbe von der seines Nachfolgers unterscheidet. "GESETZT WENN UNTERSCHIEDLICH" hat in Computerchinesisch den Namen "XOR". Anstatt die Farbe zweier Punkte abzufragen, miteinander zu vergleichen und je nach Ausgang des Vergleichs Farben zu setzen, kann

man ganz einfach jeden Punkt mit seinem Nachfolger XOR-verknüpfen, was genau denselben Effekt ergibt.

2) Da die einzelnen Zeilen unabhängig voneinander behandelt werden, zwingt uns eigentlich nichts dazu, sie hintereinander abzuarbeiten. Man kann also eine ganze SPALTE von Nachfolgepunkten (per GET... Put....6) mit einer ganzen SPALTE von Punkten verknüpfen. Diese Vorgehensweise hat zudem den Vorteil, daß sie durch den Blitter unterstützt wird (da die Befehle GET und PUT intern den LINE-A-Aufruf A007 BITBLT verwenden).

Eine solchermaßen überarbeitete Programmversion zeigt Listing 2. Auch hier erübrigt sich wohl ein Flußdiagramm, besteht es doch nur aus zwei einfachen FOR-NEXT-Schleifen. Die einzige Variable, die zusätzlich zu den in Listing 1 genannten verwendet wird, ist der String *Nachf$*, in dem jeweils eine ganze Zeile (bzw. Spalte) von Nachfolgepunkten abgelegt wird.

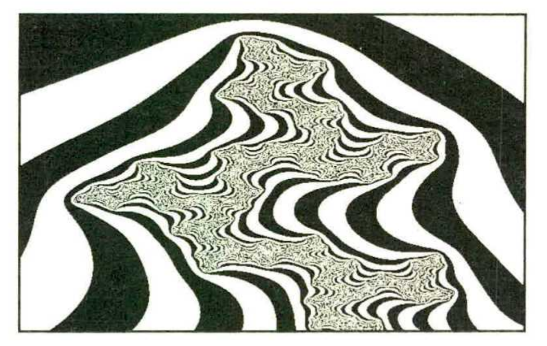


Bild 2: Vor dem Programmlauf...

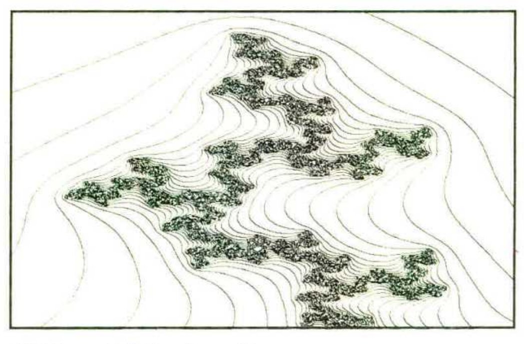


Bild 3: ... 11 Sekunden später.

Diese Ausgabe des Unfill ist nicht nur kürzer - sie ist immens schneller! Das gleiche Bild, das von Listing 1 in 18 Minuten (1080 Sekunden) bearbeitet wird, braucht mit Listing 2 nicht mal 11 Sekunden!! Damit ist die optimierte Version 100mal schneller als die nichtoptimierte, und mit Blitter vergrößert sich dieser Gewinn noch!

Die Bilder 2 und 3 zeigen zur Demonstration des Effekts ein Bild vor und nach dem Programmlauf.

Damit sind wir am Ende des Artikels angelangt. Mein Hauptanliegen war es zu zeigen, wie man sich den Einsatz teurer Hardware (Blitter z.B.) oft durch Optimierungen auf Algorithmusebene sparen kann. Wenn Sie noch Fragen, Anmerkungen oder Anregungen zum Programm haben, melden Sie sich ruhig bei mir. Meine Adresse ist:

Sven Geier
Wiesendamm 135/652
2000 Hamburg 60



```
 1:  ' ####################################
 2:  ' #  UNFILL                          #
 3:  ' #  ------                          #
 4:  ' # erste (sehr langsame) Version    #
 5:  ' # Nur zur Demonstration des Prinzips #
 6:  ' ####################################
 7:  '
 8:  xmax%=639  !Für Colorbetrieb müssen nur diese zwei
 9:  ymax%=399  !Variablen angepasst werden!
10:  '
11:  FILESELECT "\*.PI3","",a$    !Bild im DEGAS-Format
                                   auswählen
12:  OPEN "I",#1,a$  !Datei öffnen
13:  SEEK #1,34      !Header abschneiden (bei Color
                      Farben anpassen!)
14:  BGET #1,XBIOS(2),32000  !Bild in
                               Bildschirmspeicher
15:  CLOSE                    !Datei schließen
16:  SGET t$                  !Bild zwischenspeichern
17:  '
18:  FOR y%=0 TO ymax%        !Alle Zeilen einmal
                               abtasten
19:    FOR x%=0 TO xmax%-1    !Von Anfang bis Ende der
                               Zeile
20:      COLOR 0              !Punkt weiP machen...
21:      IF POINT(x%,y%)<>POINT(x%+1,y%)  !...es sei
                       denn, seine Farbe unterscheidet
22:        COLOR 1      !...sich v.d.seines Vorgängers.
23:      ENDIF          !Dann: Punkt schwarz machen!
24:      PLOT x%,y%     !und nun mit der neuen Farbe
                         plotten
25:    NEXT x%
26:  NEXT y%
27:  GET 0,0,639,399,u$  !waagerechte Abtastung
                          speichern (siehe Text)
28:  '
29:  SPUT t$        !ursprüngliches Bild wiederherstellen
30:  FOR x%=0 TO xmax%   !Alle Spalten
31:    FOR y%=0 TO ymax%-1  !Von Anfang bis Ende
32:      COLOR 0              !Wie eben, nur senkrecht...
33:      IF POINT(x%,y%)<>POINT(x%,y%+1)
34:        COLOR 1            !Farbe setzen
35:      ENDIF
36:      PLOT x%,y%           !Punkt zeichnen
37:    NEXT y%
38:  NEXT x%        !Alle Schleifen abschließen; und
                     jetzt kommts:
39:  PUT 0,0,u$,7   !waagerechte Abtastung hinein-ODER-n
40:  VOID INP(2)    !Das wars...
```

Listing 1: So funktioniert's - der Unfill-Algorithmus

```
 1:  ' ##################################
 2:  ' #  UNFILL                        #
 3:  ' #  ------                        #
 4:  ' # zweite (optimierte) Version    #
 5:  ' # Voll funktionsfähig            #
 6:  ' ##################################
 7:  '
 8:  xmax%=639  !Für Colorbetrieb müssen nur diese zwei
 9:  ymax%=399  !Variablen angepasst werden!
10:  '
11:  FILESELECT "\*.PI3","",a$ !Bild im DEGAS-Format
                                 auswählen
12:  OPEN "I",#1,a$           !Datei öffnen
13:  SEEK #1,34      !Header abschneiden
                      (bei Color Farben anpassen!)
14:  BGET #1,XBIOS(2),32000  !Bild i.Bildschirmspeicher
15:  CLOSE                    !Datei schliePen
16:  SGET t$                  !Bild zwischenspeichern
17:  '
18:  FOR y%=0 TO ymax%-1      !Alle Zeilen einmal
                               abtasten
19:    GET 0,y%+1,639,y%+1,nachf$  !Eine Zeile
                                    Nachfolgerpunkte
20:    PUT 0,y%,nachf$,6   !XOR-Verknüpfen
21:  NEXT y%          !Das war schon die erste Schleife
22:  GET 0,0,639,399,u$      !waagerechte Abtastung
                               speichern
23:  '
24:  SPUT t$        !ursprüngliches Bild wiederherstellen
25:  FOR x%=0 TO xmax%-1   !Alle Spalten
26:    GET x%+1,0,x%+1,399,nachf$  !Eine Spalte Punkte
27:    PUT x%,0,nachf$,6   !Mit XOR hineinkopieren
28:  NEXT x%               !Schleife abschließen;
                            und jetzt wieder:
29:  '
30:  PUT 0,0,u$,7          !waagerechte Abtastung
                            hinein-ODER-n
31:  VOID INP(2)           !Das wars...
```

Listing 2: ... und so geht's 100mal schneller!

ABSTRAKTE DATENTYPEN

MIT ST-PASCAL

Klaus Rindfrey

IN ST-PASCAL BESTEHT DIE MÖGLICHKEIT DES MODULAREN KOMPILIERENS, WOMIT MAN EIGENE FUNKTIONS-/PROZEDURBIBLIOTHEKEN IM OBJEKTCODE ERSTELLEN KANN. IM FOLGENDEN MÖCHTE ICH DARSTELLEN, WIE ES DAMIT SOGAR MÖGLICH IST, ABSTRAKTE DATENTYPEN, ÄHNLICH WIE "OPAQUE EXPORT" IN MODULA-2, ZU ERZEUGEN.

Zuvor möchte ich noch kurz grundsätzlich auf das modulare Programmieren mit ST-Pascal eingehen. Die etwas ungenauen Erläuterungen in den alten Handbüchern (das neue ist wirklich wesentlich besser) führten gelegentlich zu dem Mißverständnis, daß sämtliche Deklarationen in den Modulen und im Hauptprogramm genau gleich sein müßten (so z.B. in ST-Computer 12/87, S. 127, 128). Dies ist jedoch nur dann erforderlich, wenn globale Variablen des Hauptprogramms auch in Modulen benutzt werden sollen. Ansonsten kann man durchaus verschiedene Vereinbarungen in verschiedenen Modulen verwenden und so allgemein benutzbare Bibliotheken erzeugen. Außerdem überprüft der Compiler bei der Benutzung von Bibliotheksfunktionen nicht den Typ und die Anzahl der Parameter, so daß auch hier abweichende Deklarationen möglich sind.

Genau diese Tatsache kann man sich auch zur Definition abstrakter Datentypen zunutze machen. Bei abstrakten Datentypen kommt es darauf an, daß die Einzelheiten der Implementierung im Modul verborgen bleiben. Der Anwendungsprogrammierer kann nur über die Bibliotheksfunktionen mit dem abstrakten Datentyp arbeiten. Als Beispiel sei hier ein abstrakter Datentyp zur Darstellung komplexer Zahlen definiert (s. Listing 1). Wie das Modul im Hauptprogramm benutzt wird, zeigt Listing 2.

Auf die komplexen Zahlen wird nur über Adressen zugegriffen. Das entspricht genau dem Konzept von Modula-2, wo abstrakte Datentypen stets Pointertypen sind.

Wie funktioniert das nun? Wie Sie sehen, weichen die Typvereinbarungen im Hauptprogramm von denen im Bibliotheksmodul ab. Da ST-Pascal keinen Typ ADDRESS o.ä. kennt, benutze ich hier Long_Integer zur Darstellung von Adressen. Natürlich kann auch jeder andere Datentyp, der 32 Bit belegt, benutzt werden. Die Prozedur *NewComplex* stellt Speicherplatz für eine komplexe Zahl zur Verfügung, mit *DispComplex* kann dieser wieder freigegeben werden. Diese Prozeduren benutzen die Standardprozeduren New und Dispose. Wie Sie sehen, sind die Prozeduren *NewComplex* und *DispComplex* im Modul mit Pointertypen als Parameter deklariert, im Hauptprogramm dagegen mit Long_Integer. Das bereitet keine Schwierigkeiten, da Pointer nichts anderes als 32-Bit-Adressen sind. Und es ist auch notwendig, da ja die Implementierung der komplexen Zahlen im Modul verborgen bleiben soll.

Auffallen dürfte noch, daß sämtliche Prozeduren im Modul mit VAR-Parametern deklariert sind, im Hauptprogramm dagegen mit Wertparametern (außer *NewComplex* und *DispComplex*). Dies hat folgenden Grund:

Beim Wertparameter wird dem betreffenden Unterprogramm der Inhalt (Wert) des Parameters übergeben, beim VAR-Parameter nur die Adresse. Im Hauptprogramm enthalten Variablen des Typs *Complex* die Adressen komplexer Zahlen. Beim Aufruf der mit Wertparametern deklarierten Prozeduren wird nun der Wert der jeweiligen Variablen, d.h. die Adresse einer komplexen Zahl, übergeben. Da die Prozedur aber ursprünglich (im Modul) mit VAR-Parametern deklariert war, interpretiert sie diesen Wert als Adresse und arbeitet wie gewünscht mit der komplexen Zahl, die unter dieser Adresse steht.

Eine Schwäche des Konzepts soll jedoch nicht verschwiegen werden. Benutzen Sie mehrere abstrakte Datentypen

```
ADT1 = Long_Integer;
ADT2 = Long_Integer;
...
ADTn = Long_Integer;
```

in einem Programm, so betrachtet der Compiler diese alle als gleich, d.h. ADT1 bis ADTn sind alle untereinander und außerdem mit Long_Integer kompatibel. Sie müssen also als Programmierer selbst darauf achten, daß bei abstrakten Datentypen keine Typfehler vorkommen.

```
 1:  {$m+,e+  modulares Compilieren, externe Adressen.}
 2:
 3:  PROGRAM ComplexModul;
 4:    { Kleines Beispiel für eine Bibliothek mit
       komplexen Zahlen }
 5:
 6:  TYPE cxnumber = RECORD
 7:                   repart : Real; { Realteil }
 8:                   impart : Real  { Imaginärteil }
 9:                  END;
10:
11:       complex  = ^cxnumber;
12:
13:
14:   PROCEDURE NewComplex(VAR z : complex; a,b : Real)
        ;   { Reserviert speicher für z         }
15:   BEGIN
              { und setzt z <— a + i*b           }
16:     New(z);
17:     z^.repart := a; z^.impart := b
18:   END; { NewComplex }
19:
20:
21:   PROCEDURE DispComplex(VAR z : complex);
              { Speicher wieder freigeben        }
22:   BEGIN
23:     Dispose(z);
24:     z := NIL;
25:   END; { DispComplex }
26:
27:
28:   PROCEDURE Copy(VAR zq,zz : cxnumber);
               { Kopiert zq nach zz              }
29:   BEGIN
30:     zz.repart := zq.repart; zz.impart := zq.impart

31:   END; { Copy }
32:
33:
34:   FUNCTION Betrag(VAR z : cxnumber) : Real;
               { Ermittelt Betrag von z          }
35:   BEGIN
36:     Betrag := Sqrt(Sqr(z.repart)+Sqr(z.impart))
37:   END; { Betrag }
38:
39:
40:   FUNCTION Re(VAR z : cxnumber) : Real;
                { Ermittelt Realteil von z       }
41:   BEGIN Re := z.repart                                →
```

```
42:    END; { Re }
43:
44:
45:    FUNCTION Im(VAR z : cxnumber) : Real;
                 { Ermittelt Imaginärteil von z }
46:    BEGIN Im := z.impart
47:    END; { Im }
48:
49:
50:    FUNCTION Equal(VAR z1,z2 : cxnumber) : Boolean;
                 { Prüft, ob z1 = z2               }
51:    BEGIN
52:      Equal := (z1.impart = z2.impart) And (z1.repart
              = z2.repart)
53:    END; { Equal }
54:
55:
56:    PROCEDURE Add(VAR s1,s2,erg : cxnumber);
                 { Komplexe Addition               }
57:    BEGIN
58:      erg.repart := s1.repart+s2.repart;
59:      erg.impart := s1.impart+s2.impart
60:    END; { Add }
61:
62:
63:    PROCEDURE Mult(VAR f1,f2,prd : cxnumber);
                 { Komplexe Multiplikation         }
64:    VAR f1r,f1i,f2r,f2i : Real;
65:    BEGIN
66:      f1r := f1.repart; f1i := f1.impart;
67:      f2r := f2.repart; f2i := f2.impart;
68:      prd.repart := f1r*f2r-f1i*f2i;
69:      prd.impart := f1r*f2i+f1i*f2r
70:    END; { Mult }
71:
72:
73:    PROCEDURE Divi(VAR z,n,q : cxnumber);
                   { Komplexe Division             }
74:    VAR bq,zr,zi,nr,ni : Real;
75:    BEGIN
76:      zr := z.repart; zi := z.impart;
77:      nr := n.repart; ni := n.impart;
78:      bq := Sqr(nr)+Sqr(ni);
79:      q.repart := (zr*nr+zi*ni)/bq;
80:      q.impart := (nr*zi-zr*ni)/bq;
81:    END; { Divi }
82:
83:  BEGIN END.
```

```
 1:  PROGRAM ComplexTest;
 2:
 3:  TYPE complex = Long_Integer; { Typ complex muP 32
                                    Bit groP sein }
 4:
 5:  VAR z1,z2,z3 : complex;
 6:
 7:
 8:  PROCEDURE NewComplex(VAR z : complex; a,b : Real);
           EXTERNAL;
 9:
10:  PROCEDURE DispComplex(VAR z : complex);
           EXTERNAL;
11:
12:  PROCEDURE Copy(zq,zz : complex);
           EXTERNAL;
13:
14:  FUNCTION Betrag(z : complex) : Real;
           EXTERNAL;
15:
16:  FUNCTION Re(z : complex) : Real;
           EXTERNAL;
17:
18:  FUNCTION Im(z : complex) : Real;
           EXTERNAL;
19:
20:  FUNCTION Equal(z1,z2 : complex) : Boolean;
           EXTERNAL;
21:
22:  PROCEDURE Add(s1,s2,erg : complex);
           EXTERNAL;
23:
24:  PROCEDURE Mult(f1,f2,prd : complex);                 →
```

```
             EXTERNAL;
25:
26:  PROCEDURE WriteComplex(z : complex);
27:  BEGIN
28:     Write(Re(z):5:3, ' + ', Im(z):5:3, ' * i')
29:  END; { WriteComplex }
30:
31:
32:  BEGIN
33:    NewComplex(z1,1,2);           { z1 = 1 + 2i }
34:    NewComplex(z2,0,0);           { z2 = 0      }
35:    NewComplex(z3,0,1);           { z3 = i      }
36:    Copy(z1,z2);
37:    Writeln(' z1 —> z2  : ');
38:    Write(' z2 = ');
39:    WriteComplex(z2);
40:    Writeln; Writeln;
41:    Write(' Betrag(' );
42:    WriteComplex(z1);
43:    Write(') = ', Betrag(z1));
44:    Writeln; Writeln;
45:    Add(z2,z3,z1);
46:    Write(' z2+z3 = ');
47:    WriteComplex(z1); Writeln;
48:    Mult(z2,z3,z1);
49:    Write(' z2*z3 = ');
50:    WriteComplex(z1); Writeln;
51:    DispComplex(z1);
52:    DispComplex(z2);
53:    DispComplex(z3);
54:    Writeln('Mit Return zurück zum Desktop');
55:    Readln;  { Auf Return warten }
56:  END.
```

ST-Ecke

LINE-A

Von Mäusen und Elfen

Zugegeben, die Routinen wie beispielsweise die Textausgabe oder BitBlt des LINE-A haben sehr viele Parameter. Trotzdem wollen wir uns dadurch nicht abschrecken lassen. In dieser ST-Ecke werden wir gleich mehrere Routinen des Line-A beschreiben, so daß Sie am Ende in der Lage sein werden, die Maus zu manipulieren sowie die Sprites vernünftig einzusetzen.

Schlimm, Schlimm...

... ist es, daß ich leider (hoffentlich zum letzten Mal) drei kleine Fehler zugeben muß, die ich Sie bitte, in Ihren abgetippten Listings zu berichtigen. Dabei handelt es sich um die Listings von *set_line_a()*, *set_line_color()* und *a008()*, die Sie nochmals in Listing 7 finden werden, wobei die nicht korrekten Zeilen mit Pfeil im Kommentar bezeichnet sind. In *set_line_a()* war die Berechnung der Bytes pro Zeile bei angegebener Plane-Zahl nicht richtig: Die monochrome Auflösung hat 80, die beiden anderen 160 Bytes pro Zeile. Der Fehler in *set_line_color* ist schon fast peinlich. Hier sollten die einzelnen Line A-Variablen *COLBIT_1* bis *COLBIT_3* auf true sein, falls die Bits 1 bis 3 gesetzt sind. Dabei auf 1 und nicht auf die Wertigkeit der Bits (2, 4, 8) abzufragen, ist natürlich Blödsinn, so daß keine unterschiedlichen Farben erscheinen. Den endgültig letzten Fehler finden Sie in *a008()*, wo die Übergabevariable das auszugebende Zeichen als 'signed char' übernimmt, was falsch ist, da alle Zeichen über 127 zu undefinierten Reaktionen führen - Zeichen sollten prinzipiell immer unsigned definiert werden. Eine kleine Ergänzung zum Listing der Routine *a005*: Im Binding wird die Konstantendefinition *LA_DFR* benutzt, die Sie besser gegen 0xa005 austauschen sollten.

Los geht's

Sicherlich haben Sie sich mit dem Grafikkonzept des ATARI ST ein wenig auseinandergesetzt. Dabei ist nicht zu übersehen, daß die Maus als Zeiger zum Auswählen oder Zeichnen großzügig benutzt wird. Falls Sie schon einmal Grafik programmiert haben, ist Ihnen sicherlich schon einmal aufgefallen, daß es wichtig ist, die Maus vorher auszuschalten, denn sie ist kein Hardwaresprite wie Sie es vielleicht von Zeiten des Commodore 64 oder ATARI 400/800/1200(XL) her kennen. Das heißt, daß die Maus eine ganz normale Grafik ist, die, bevor sie gezeichnet wird, ihren Hintergrund retten und bei einer Bewegung wieder restaurieren muß. Da das GEM, wie schon zu Beginn unserer Serie erwähnt, im Endeffekt auf den Kernel des ST und damit das Line-A zugreift, muß auch Line-A in der Lage sein, die Maus zu manipulieren. Allerdings stellt hier das Line-A nur drei Routinen zur Verfügung, die es ermöglichen, die Maus ein- und auszuschalten sowie eine Änderung der Mausform vorzunehmen. Das Setzen wie auch das Abfragen der Mausposition ist über Line-A-Routinen nicht möglich. Im Gegensatz zu den letzten beiden Routinen BitBlt und Textausgabe sind diese Routinen sehr einfach zu handhaben. Die Routine *A00A* dient dem Ausschalten und *A009* dem Einschalten der Maus. Wenn Sie Line-A-A009 aufrufen, erscheint die Maus auf dem Bildschirm, und gleichzeitig wird der Hintergrund der Maus abgespeichert. Wollen Sie nun irgendwelche Grafiken zeichnen, müssen Sie erst die Maus mit A00A ausschalten. Diese Routine holt sich aus ihrem Puffer den zwischengespeicherten Hintergrund und restauriert ihn wieder. Auch wenn Line-A keine Koordinatenmanipulation zur Verfügung stellt, wird natürlich immer der aktuelle

Maushintergrund im Puffer restauriert. Da diese Routinen schlecht mit einer Demo darzustellen sind, finden Sie in Listing 2 nur die eigentlichen Bindings. Interessant ist allerdings: Vielleicht haben Sie schon unter GEM festgestellt, daß Sie die Maus so oft wieder einschalten müssen, wie Sie sie auch ausgeschaltet haben. Das liegt daran, daß intern ein Zähler bei A00A hoch- und bei A009 heruntergezählt wird. Erst wenn der Zähler Null erreicht, erscheint die Maus wieder 'am Horizont'. Dieses Mitzählen läßt sich allerdings ausschalten, so daß die Routinen sofort ihre Wirkung zeigen. Dies ist auch der einzige Parameter, den man A009/A übergeben muß: Er ist größer Null für die zählende Maus und gleich 0 für deren sofortige Wirkung. In den Beispielen der letzten Folgen habe ich die Routinen benutzt, so daß sie dort bei Bedarf schauen können.

Maus-Show

Wie schon erwähnt, kann man das Aussehen der Maus verändern. Die Mausform wird aufgrund einer bestimmten Struktur definiert, die Sie in Listing 4 oder in der Include-Datei 'gemdefs.h' finden sollten. Fangen wir oben an: Die ersten beiden Einträge definieren den sogenannten Hot-Spot der Maus, der ihren empfindlichen Teil definiert. Sicherlich ist es nicht sinnvoll, bestimmte Reaktionen auf jeden Punkt der Maus zu beziehen, sondern einen speziellen dafür auszusuchen. So ist der Hot-Spot bei dem bekannten Pfeilbild an der Spitze des Pfeils oder bei einer Hand am Ende des zeigenden Fingers. Ich habe Ihnen als kleine Überraschung ein Mausbild definiert, welches ein kleines Mäuschen darstellen soll. Dabei habe ich den Hot-Spot auf die Schnauze der Maus gesetzt. Der Hot-Spot wird als relative Koordinate bezogen auf die linke obere Ecke der Maus angegeben. Der nächste Eintrag legt die Anzahl der Planes einer Maus fest, aber leider wird sowieso nur eine Plane unterstützt, und deshalb wird dieser Parameter auf eins gesetzt. Auch wenn die Maus dadurch nur einfarbig sein darf, gibt es die Möglichkeit Daten- und Maskenfarbe mit *mf_fg* und *mf_bg* festzulegen. Auch wenn die Parameter als Vorder- und Hintergrundfarbe übersetzt werden können, geben Sie tatsächlich Daten- und Maskenfarbe an, denn den Hintergrund ändern die Mausroutinen nicht.

```
 1: #include <line_a.h>
 2: #include <gemdefs.h>
 3: #include <stdio.h>
 4: #include <osbind.h>
 5:
 6:
 7: #define BIG_CHAR 16 /* größtes Zeichen, sollte
                        mind. max (s. set_line_font) */
 8:                     /* aller Zeichensätze sein */
 9:
10: #define void /**/   /* Nur definieren, falls nicht bekannt */
11:
12: #ifndef abs         /* Makro für Absolutbetrag */
13:   #define abs(x) ((x<0) ? x*-1:x)
14: #endif
15:
16:
17: /* Folgende Strukturdefinitionen nur einbinden,
18:    falls noch nicht vorhanden */
19:
20: /*
21: typedef struct mfstr
22: {
23:       int     mf_xhot;
24:       int     mf_yhot;
25:       int     mf_nplanes;
26:       int     mf_fg;
27:       int     mf_bg;
28:       int     mf_mask[16];
29:       int     mf_data[16];
30: } MFORM;
31:
32:
33: typedef struct sfstr
34: {
35:       int xhot;
36:       int yhot;
37:       int form;
38:       int bgcol;
39:       int fgcol;
40:       int image[32];
41: }SDB;
42: */
43:
44: LINE_A *linea, *a000();  /* Zeiger auf Line-A-Variablen */
45: long *fonts, *routines;  /* Zeiger auf Routinen und Fonts */
46:
47: int scratch_buf[BIG_CHAR*2];      /* Puffer für Texteffekte */
48: MFDB bildschirm={0L,640,400,40,0,1,0,0,0};   /* Monochrom */
49:
50: int fill[]={                /* Füllmuster, 16 Zeilen hoch */
51: 0x2E74,0xA5A5,0x300C,0x9C39,
52: 0xCFF3,0x4812,0x6426,0xD3CB,
53: 0xD3CB,0x6426,0x4812,0xCFF3,
54: 0x9C39,0x300C,0xA5A5,0x2E74};
55:
56: /* Ein Ufo-Sprite */
57: int sprite_array[]={0x0000,0x0000,0x0001,0x0002,0x0005,
58:                     0x0000,0x0000,0x0FF0,0x0000,
59:                     0x1FF8,0x0FF0,0x3FFC,0x1FF8,
60:                     0x7FFE,0x3FFC,0xFFFF,0x7FFE,
61:                     0x4002,0x2AAC,0x3FFC,0x1FF8,
62:                     0x1FF8,0x0FF0,0x0FF0,0x0000,
63:                     0x0000,0x0000,0x0000,0x0000,
64:                     0x0000,0x0000,0x0000,0x0000,
65:                     0x0000,0x0000,0x0000,0x0000};
66:
67: /* Das gleiche Sprite als Struktur definiert */
68:
69: SDB sprite= { 0x0000,0x0000,0x0001,0x0002,0x0005,
70:             { 0x0000,0x0000,0x0FF0,0x0000,
71:               0x1FF8,0x0FF0,0x3FFC,0x1FF8,
72:               0x7FFE,0x3FFC,0xFFFF,0x7FFE,
73:               0x4002,0x2AAC,0x3FFC,0x1FF8,
74:               0x1FF8,0x0FF0,0x0FF0,0x0000,
75:               0x0000,0x0000,0x0000,0x0000,
76:               0x0000,0x0000,0x0000,0x0000,
77:               0x0000,0x0000,0x0000,0x0000 }
78:             };
79:
80: /* Eine Maus-Maus */
81: int mouse[]= { 0x0000,0x0008,0x0001,0x0000,0x0001,
82:                0x0000,0x0030,0x0038,0x0FFC,
83:                0x1FFC,0x3FFC,0x6FFC,0xFFFC,
84:                0xFFF8,0xFFF8,0x0000,0x0000,
85:                0x0000,0x0000,0x0000,0x0000,
```

→

Maskerade

Die Größe der Maus ist auf 16x16 fest definiert und kann daher nicht beliebig erweitert werden. Dabei wird eine Mauszeile (16 Punkte) in einem Integerwort abgespeichert, wodurch sich die Tatsache ergibt, daß jedes Bit einem gesetzten Punkt entspricht. Das höchstwertigste Bit entspricht dem linken und das niederwertigste dem rechten Punkt. Die nächsten 15 Zeilen der Maus findet man in den folgenden 15 Integer-Worten. Diese sich ergebende Datenmenge von 16 Integer-Worten werden Mausformdaten genannt und finden sich am Ende der Mausform unter dem Namen *mf_data*. Bliebe eigentlich nur noch *mf_mask* übrig: Diese Maske ist genauso groß wie die Datenmenge, so daß ein Punkt der Maske dem entsprechenden in den Daten zugeordnet ist. Nun besteht folgende Beziehung: Ist ein Punkt in den Daten gesetzt, erscheint auf jeden Fall an dieser Stelle dieser Punkt in der Maus. Ist aber an einer bestimmten Stelle ein Punkt nicht gesetzt, stellt man sich die Frage, ob der Hintergund des Bildes, über dem die Maus sich befindet, oder einfach die gewählte Hintergrundfarbe erscheinen soll? Falls ein Punkt in den Daten nicht, der zugehörige Punkt in den Maske aber gesetzt ist, erscheint an dieser Stelle die gewählte Hintergrundfarbe, während das Bild durchscheint, falls in Daten und Maske der Punkt nicht gesetzt ist. Dies kann man nun für zwei Dinge nutzen. Zunächst einmal ist die häufigste Nutzungsart das Zeichnen eines Randes um die eigentliche Maus. Dabei zeichnet man die Maus höchstens 14*14, um seitlich Platz für einen Rand zu lassen, der allerdings nur in der Maske zu finden ist. Dieser Rand wirkt dann als Umrandung, wodurch die Maus auf vielen Hintergründen sehr gut zu erkennen ist. Eine weitere Anwendung findet sich, in dem man Bereiche innerhalb der Maus durchsichtig erscheinen läßt. Wie schon erwähnt, habe ich Ihnen eine Maus als Mausform definiert (Listing 1). Diese Maus hat ein kleines Auge, das durchsichtig ist - probieren Sie es einfach mal aus, indem Sie die Maus ein wenig über den (nicht gerade weißen) Hintergrund schieben.

Das Binding für die Transform-Mouse-Routine *A00B* ist relativ einfach. Die Mausformstruktur ist 2*2*16+10 Bytes lang, die allesamt in das INTIN[]-Feld geschrieben werden müssen, was eigentlich ein Kopieren der Werte bedeutete. Wir können die Arbeitszeit auch rationalisieren, indem wir dem Line-A für einen

```
 86:                  0x0000,0x0000,0x0010,0x0028,
 87:                  0x0F88,0x1FC8,0x2FF8,0x7FF8,
 88:                  0xFFF0,0x1040,0x0000,0x0000,
 89:                  0x0000,0x0000,0x0000,0x0000} ;
 90:
 91:  /* Maus als Struktur: Sinnvollere Art, die Maus zu definieren */
 92:  MFORM maus=  { 0x0000,0x0008,0x0001,0x0000,0x0001,
 93:                  { 0x0000,0x0030,0x0038,0x0FFC,
 94:                    0x1FFC,0x3FFC,0x6FFC,0xFFFC,
 95:                    0xFFF8,0xFFF8,0x0000,0x0000,
 96:                    0x0000,0x0000,0x0000,0x0000 },
 97:                  { 0x0000,0x0000,0x0010,0x0028,
 98:                    0x0F88,0x1FC8,0x2FF8,0x7FF8,
 99:                    0xFFF0,0x1040,0x0000,0x0000,
100:                    0x0000,0x0000,0x0000,0x0000},
101:               };
102:
103:  main()
104:  {
105:     /* Hier kommen die unter AUFRUF aufgeführten Aufrufe hinein */
106:  }
```

Listing 1: Der Hauptteil für unsere Demos

```
Eingabe: INTIN[0]: Mauszählung (0=abgeschaltet
                                >0=eingeschaltet)

Ausgabe: keine

Routine:

  1:  a009(arg)
  2:  int arg;                    /* 0: internen Zähler miPachten */
  3:  {
  4:     linea->INTIN[0]=arg;     /* Verschachtelungspar. */
  5:     asm{
  6:        dc.w 0xa009           /* Show_mouse */
  7:        }
  8:  }

Aufruf:

a009(1);   /* Maus (eventuell) einschalten */
a009(0);   /* Maus sofort einschalten */
```

Listing 2: Einschalten der Maus - A009

```
Eingabe: INTIN[0]: Mauszählung (0=abgeschaltet
                                >0=eingeschaltet)

Ausgabe: keine

Routine:

  1:  a00a(arg)
  2:  int arg;  /* 0: internen Zähler miPachten */
  3:  {
  4:     linea->INTIN[0]=arg; /* Verschachtelungspar. */
  5:     asm{
  6:        dc.w 0xa00a       /* Hide_mouse */
  7:        }
  8:  }

Aufruf:

a00a(1);   /* Maus (eventuell) ausschalten */
a00a(0);   /* Maus sofort einschalten */
```

Listing 3: Ausschalten der Maus - A00A

```
Eingabe: INTIN[]: Werte der Mausstruktur:
           typedef struct mfstr
              {
                 int      mf_xhot;       /* Hot-Spot X */
                 int      mf_yhot;       /* Hot-Spot Y */
                 int      mf_nplanes;    /* Anzahl der Planes = 1 */
                 int      mf_fg;         /* Vordergrundfarbe */
                 int      mf_bg;         /* Hintergrundfarbe */
                 int      mf_mask[16];   /* Maske */
                 int      mf_data[16];   /* Daten */
```

→

Geht es um den DIN A4

SPAT SCANNER

EXTRAS:

REPARATUREN
THERMOPAPIER
PIC READER
SPAT SOFTWARE 3.9
SPAT TEXTER
SPAT UTILITY
SPAT SCHRIFTEN
SPAT BILDERALBEN
SPAT LABOR FÜR
MEGA ST 2 UND 4

SOFTWARE UPDATES

RAINER MERTENS
MIERENDORFFPLATZ 8
1000 BERLIN 10
TEL. 030 345 30 61
FAX. 030 345 81 72

982.-DM

H&T
HARDWARE & TECHNIK

INFO KOSTENLOS ANFORDERN
STÜTZPUNKT-HÄNDLER GESUCHT
LIEFERUNG DIREKT AB LAGER
HOTLINE SERVICE 15-16 UHR
ARTIKEL ST-COMPUTER 5/89



Moment vortäuschen, daß die Mausformstruktur das INTIN[]-Feld sei. Das erreichen wir dadurch (Listing 4), daß wir zunächst den echten Zeigerwert zwischenspeichern, die Mausformadresse in den INTIN[]-Zeiger schreiben, A00B aufrufen und danach den echten INTIN-Zeiger für spätere Aufrufe zurückschreiben.

```
            } MFORM;

Ausgabe: keine

Routine:

  1:  a00b(mouse_array)
  2:  int *mouse_array;
  3:  {
  4:      int *merke;
  5:
  6:      merke=linea->INTIN;                /* Intin-Zeiger retten */
  7:      linea->INTIN=(int*)mouse_array;    /* Maus-Feld 'ist jetzt INTIN' */
  8:
  9:      asm{
 10:          dc.w 0xa00b                    /* Mausform ändern */
 11:          }
 12:      linea->INTIN=merke;                /* Intin-Zeiger restaurieren */
 13:  }

Aufruf:

a00b(mouse);                         /* Maus als Integer-Array definiert */
a00b(&maus);                         /* Maus als Struktur definiert */
```

Listing 4: Mausform ändern - A00B

Von Fabelgestalten und Elfen

Was hat eine Elfe oder eine Fabelgestalt mit Line-A zu tun, werden Sie sich fragen. Ganz einfach: Das Wort Sprite bedeutet soviel wie Elfe oder kleine Gestalt, die anderen Gutes oder Böses zufügt. Der Begriff hat sich sicherlich daraus ergeben, daß vor einigen Jahren mit Sprites hauptsächlich kleine Figuren in Spielen realisiert wurden. Wie oben angesprochen, verfügen Homecomputer oft über hardwareseitig unterstützte Sprites, das heißt, daß die Sprites, sind sie einmal definiert, nur noch angeschaltet und die Koordinaten verändert werden müssen. Selbst Kollisionen zwischen Sprites oder Vorrangeigenschaften werden ausgewertet. Das Beste aber an diesem Konzept ist, daß diese Hardwaresprites die eigentliche Hintergrundgrafik nicht zerstören, da Sie völlig unabhängig sind. Nach einer solchen Schwärmerei kann man nur sagen: Schade, daß es sowas beim ST nicht gibt. Die Routinen, die beim ST Sprite-Routinen genannt worden sind, haben nur die Aufgabe, ein Objekt zu zeichnen und dabei die vorherige Position wieder zu restaurieren. Leider sind diese Objekte auch nur 16x16 groß und lassen nur eine Plane (gleich 2 Farben) zu - das Ganze riecht stark nach Mauszeichenroutine, da sie aufgrund des Zeichenparameters form sehr nach der GEM-Mausroutine ausschaut.

```
Eingabe: A2: Zeiger auf Sprite-Save-Block
             (10+64*Planes)

Ausgabe: Register A6 wird zerstört

Routine:

  1:  void a00c(buf)  /* Lösche Sprite */
  2:  int *buf;
  3:  {
  4:      asm
  5:      {
  6:          movem.l A2/A6,-(A7)  /* Register retten */
  7:          move.l  buf(A6), A2  /* Sprite-Hintergrundsblock */
  8:          dc.w    0xa00c       /* Sprite löschen */
  9:          movem.l (A7)+, A2/A6 /* Register restaurieren */
 10:      }
 11:  }

Aufruf:

siehe Demonstration für A00D
```

Listing 5: Löschen eines Sprites - A00C

Beweglich

Im Gegensatz zu der Mausroutine ist die Spriteroutine in der Lage, das gewünschte Bild an eine bestimmte Koordinate zu zeichnen. Die X-Koordinate wird in Register D0 und die Y-Koordinate in Register D1 geschrieben. In Register A0 schreiben wir die Adresse unseres Sprite-Definitionsblocks, der so ähnlich aussieht, wie die MFORM-Struktur einer Maus. Die ersten beiden Einträge geben die relative X-Koordinate des Sprites bezogen auf die angegebene X/Y-Koordinate an. Der dritte Parameter gibt an, ob die Sprites im sogenannte VDI-Modus gezeichnet oder die Daten mit dem Hintergrund Exklusiv Oder (XOR)-verknüpft werden sollen, sofern kein Maskenpunkt gesetzt wird. Am besten läßt sich dies an Tabelle 1 zeigen.

Die nächsten beiden Parameter *bgcol* und *fgcol* geben an, in welcher Farbe die Punkte der Maske oder der Daten gezeichnet werden sollen. Darauf folgen die Daten des Sprites, die aber anders angeordnet sind als die der Maus. Während bei der Maus zunächst die Maske und dann die Mausdaten kommen, wechseln sich bei Sprites immer eine Datenzeile mit einer Maskenzeile ab. Daraus ergibt sich die Reihenfolge *Datenwort0, Maskenwort0, Datenwort1, Maskenwort1* und so weiter.

Beim Binding werden im Gegensatz zu

Daten	Maske	VDI-Ergebnis	XOR-Ergebnis
0	0	Hintergrund	Hintergrund
0	1	Maskenfarbe	Maskenfarbe
1	0	Datenfarbe	Daten-XOR-Hintergrund
1	1	Datenfarbe	Datenfarbe

Tabelle 1: VDI- und XOR-Ergebnisse beim Zeichnen eines Sprites

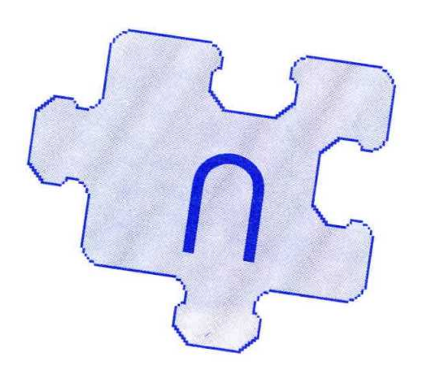
den meisten anderen Line A-Routinen die Parameter in den Registern D0, D1, A0 und A2 übergeben. In D0 und D1 steht die Koordinate, an der das Sprite gezeichnet werden soll; A0 enthält einen Zeiger auf den Sprite-Definitionsblock und A2 einen auf einen Speicherplatz, in dem LINE-A den Hintergrund des Sprites speichern kann. Daher muß der Puffer auch 10 Bytes (für Sprite-Parameter) +64 Bytes pro Farbebene Platz haben.

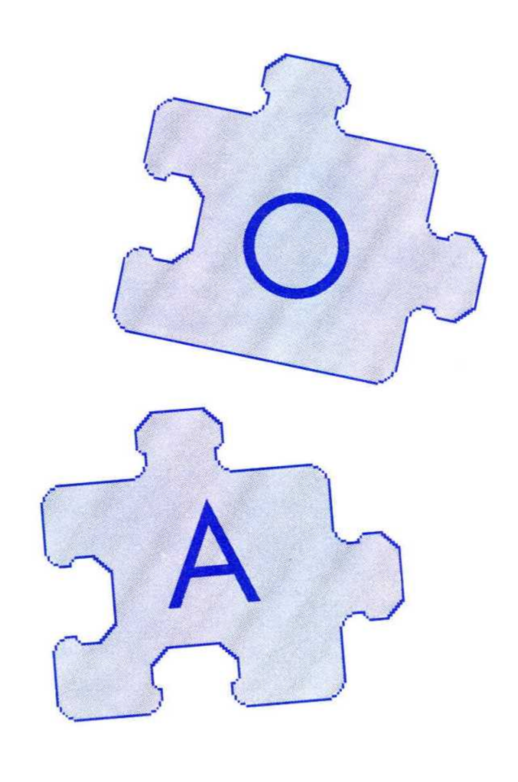
Undraw Sprite

Natürlich gibt es zum Löschen des Sprites eine Routine, die unter dem Wert *A00C* angesprochen wird. Sie erwartet nur den Zeiger des Hintergrundpuffers des entsprechenden Sprites, damit der Bildschirm restauriert werden kann. Es sicherlich klar, daß jedes Sprite seinen eigenen Hintergrundpuffer benötigt.

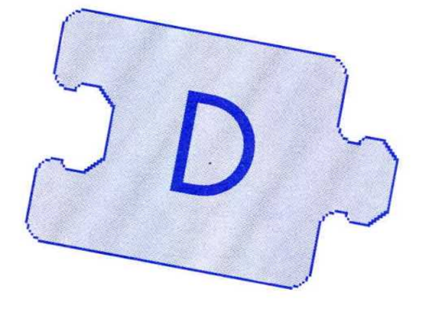
SH

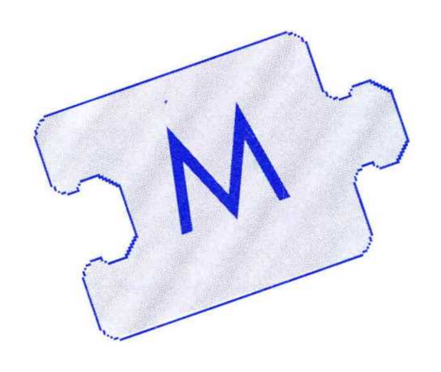






MODULA-2 KURS





Willkommen zum siebten Teil der Modula-2 Serie, in der es endlich um Module geht. Sie haben der Sprache den Namen gegeben und sind Ihnen bei fast allen Beispielprogrammen durch die Importe begegnet. In dieser Folge lernen Sie, lokale und externe Module zu benutzen und zu schreiben.

Sichtbarkeit und Lebensdauer

Das Modulkonzept bringt eine genaue Beschreibung von Sichtbarkeit und Lebensdauer, Schnittstellenüberprüfung und getrennte Compilierung. Doch der Reihe nach.

Alle Objekt eines Programmes, also z.B. Variablen, Prozeduren oder Typdefinitionen, haben zwei besondere Eigenschaften: die Sichtbarkeit und die Lebensdauer.

Lokale Variablen zum Beispiel können nur innerhalb der sie umschließenden Prozedur verwendet werden, da sie nur in diesem Bereich bekannt sind. Außerhalb einer solchen Prozedur ist die Variable nicht sichtbar. Der Sichtbarkeitsbereich einer lokalen Variablen ist also hier die Prozedur.

Gleichzeitig würde eine globale Variable gleichen Namens von der lokalen Variablen in der Prozedur überlagert. Der Sichtbarkeitsbereich einer globalen Variablen ist also das gesamte Programm abzüglich der Prozeduren, in denen eine lokale mit gleichem Bezeichner definiert ist.

Innerhalb ihres Sichtbarkeitsbereiches kann der Bezeichner eines Objektes - das Variablenbeispiel kann analog auch z.B. auf Prozeduren bezogen werden - im Programmtext verwendet werden. Außerhalb ist der Name dem Compiler nicht bekannt - nicht sichtbar - oder bezeichnet ein anderes Objekt. Diese Sichtbarkeitsregelung ist streng hierarchisch, da es in einem Sichtbarkeitsbereich keine doppelt verwendeten Bezeichner geben kann. Lediglich die Überlagerung durch lokale Bezeichner ermöglicht dies.

Weiterhin von Wichtigkeit ist die Lebensdauer eines Objekts. Bei der Beschreibung der lokalen Variablen haben Sie gelernt, daß sie beim Eintritt in eine Prozedur eingerichtet und beim Verlassen wieder entfernt werden. Dabei geht ihr Inhalt verloren und kann bei einem erneuten Aufruf einer Prozedur nicht mehr erreicht werden.

Die Lebensdauer einer lokalen Variablen entspricht genau dem Zeitraum, in dem der Rechner in der Prozedur arbeitet. Bei globalen Variablen ist die Lebensdauer die Laufzeit des gesamten Programms.

Die Sprache legt also genaue Regeln für Sichtbarkeit und Lebensdauer fest. Es kann Probleme geben, in denen der Programmierer diese Regeln vielleicht selber bestimmen will. Und dafür kann er in Modula Module verwenden.

Lokale Module

Nehmen wir als Beispiel eine ganz einfache Aufgabe: Im Programm - nehmen wir an, es sei eine Simulation - soll eine Art "Uhr" vorkommen, die durch einen Prozeduraufruf jeweils um einen Zeitwert weitergezählt werden soll. Eine weitere Prozedur soll den Stand der Uhr auf dem Bildschirm ausgeben.

Eine Lösung, bei der das Weiterzählen so aussieht

```
PROCEDURE ZaehleWeiter;
VAR Uhrzeit:INTEGER;
BEGIN
  Uhrzeit:=Uhrzeit+1;
END ZaehleWeiter;
```

macht keinen Sinn, denn die Lebensdauer

Anworten von Teil VI

1. Die rekursive Prozedur könnte so aussehen (s. Listing rechts).

```
PROCEDURE Summe(Feld:ARRAY OF CHAR; Startindex:INTEGER):LONGINT;
BEGIN
 IF Startindex<=HIGH(Feld) THEN
   RETURN Summe(Feld,Startindex+1)+LONGINT(Feld[Startindex]);
 END;
END Summe;
```

Dabei ist die Annahme, daß Summe alle Feldelemente ab Startindex aufaddiert. Summe wird nun so programmiert, daß das Ergebnis das Element mit Index Startindex plus die Summe aller weiteren Felder ist. Diese Summe errechnet der Selbstaufruf mit erhöhtem Startindex. Die Abbruchbedingung beendet die Rekursion, wenn der Startindex die Größe des Feldes überschreitet.

2. Puzzle liefert die folgende Ausgabe (Bildschirmausgaben fett):

```
P(1)
|                                                  |
1  P(0+2)                                          0
   |                                              /
   2  P(1+2)                                     1
      |                                         /
      3 P(2+2)                                 2
        |                                     /
        4 P(3+2)                             3
          |                                 /
          5 P(4+2)                         4
            |                             /
            6 P(5+2)                     5
              |                         /
              7 P(6+2)                 6
                |                     /
                8 P(7+2)             7
                  |                 /
                  9 P(8+2) 8
                    |     /
                    10 - 9
```

Die Darstellung zeigt, wie sich die Rekursion Stufe um Stufe hinabarbeitet. Beim Aufstieg wird jeweils das vorher um 1 verminderte n ausgegeben, das auf jeder Rekursionsstufe lokal neu eingerichtet ist.

3. Die Lösung ist recht einfach, da Sie sich an den Beispielen in der sechsten Folge orientieren konnten. Für jede Ampel wird eine eigene Menge eingerichtet. Mit zwei Mengentypen wird zwischen der drei- und der zweifarbigen Ampel unterschieden. Dabei dürfen die Bezeichner der Mengenelemente nicht doppelt in der Deklaration der Aufzählungstypen vergeben werden. Das "Schalten" der Ampeln geschieht mit Mengenoperatoren oder den Standardfunktionen INCL und EXCL.

```
    AutoAmpel : SET OF AutoFarben;
    FussFarben: (FRot, FGruen);
    FussAmpel : SET IF FussFarben;
VAR ch   : CHAR;
   Auto : AutoAmpel;

PROCEDURE ShowCar(Ampel: AutoAmpel);
BEGIN
 WriteString('Auto-Ampel: ');
 IF Rot   IN Ampel THEN WriteString('Rot ')   END;
 IF Gelb  IN Ampel THEN WriteString('Gelb ')  END;
 IF Gruen IN Ampel THEN WriteString('Gruen ') END;
 WriteLn;
END ShowCar;
PROCEDURE ShowWalker(Ampel: FussAmpel);
BEGIN
 WriteString('Fußgänger-Ampel: ');
 IF FRot   IN Ampel THEN WriteString('Rot ')   END;
 IF FGruen IN Ampel THEN WriteString('Gruen ') END;
 WriteLn;
END ShowWalker;
BEGIN
 Auto:=AutoAmpel{Rot};  Fuss:=FussAmpel{FRot};
 ShowCar; ShowWalker; Read(ch);
 INCL(Auto,Gelb);    (* Oder: Auto:=AutoAmpel{Rot,Gelb} *)
 ShowCar; ShowWalker; Read(ch);
 EXCL(Auto,Rot);  EXCL(Auto,Gelb);
 INCL(Auto,Gruen);   (* Oder: Auto:=AutoAmpel{Gruen}    *)
 ShowCar; ShowWalker; Read(ch);
 EXCL(Auto,Gruen);
 INCL(Auto,Gelb);    (* Oder: Auto:=AutoAmpel{Gelb}     *)
 ShowCar; ShowWalker; Read(ch);
 EXCL(Auto,Gelb);
 INCL(Auto,Rot);     (* Oder: Auto:=AutoAmpel{Rot}      *)
 ShowCar; ShowWalker; Read(ch);
 EXCL(Fuss, FRot);
 INCL(Fuss, FGruen); (* Oder: Fuss:=FussAmpel{FGruen}   *)
 ShowCar; ShowWalker; Read(ch);
 EXCL(Fuss, FGruen);
 INCL(Fuss, FRot);   (* Oder: Fuss:=FussAmpel{FRot}     *)
 ShowCar; ShowWalker; Read(ch);
END Ampeln;
```

4. Um die Menge der erlaubten Zeichen übergeben zu können, muß zunächst ein entsprechender Typ deklariert werden. Die Prozedur erhält einen weiteren Parameter und der Test, ob eine Eingabe erlaubt ist, geschieht mit dem IN-Operator. Die Rückgabe als Großbuchstabe ist mit der Standardfunktion CAP einfach zu realisieren.

```
FROM InOut IMPORT WriteString, Read;
...
TYPE  SETOFCHAR = SET OF CHAR ;
...
PROCEDURE Meldung(Text:ARRAY OF CHAR; Erlaubt: SETOFCHAR):CHAR;
BEGIN
 WriteString(Text);
 REPEAT
  Read(ch);
 UNTIL (ch IN Erlaubt);
 RETURN CAP(ch);
END Meldung;
```

von Uhrzeit ist auf die Abarbeitung von ZaehleWeiter beschränkt. Damit wird zwar irgendetwas weitergezählt, aber sicherlich nicht eine simulierte Uhr. Da Uhrzeit zudem nur in ZaehleWeiter sichtbar ist, kann eine andere Prozedur darauf nicht zugreifen und sie folglich auch nicht ausgeben.

Ein zweiter Lösungsansatz: Die momentane Uhrzeit wird als Parameter übergeben:

```
PROCEDURE ZaehleWeiter(VAR Uhrzeit:INTEGER);
BEGIN
  Uhrzeit:=Uhrzeit+1;
END ZaehleWeiter;

PROCEDURE SchreibeUhrzeit(Uhrzeit:INTEGER);
BEGIN
  WriteInt(Uhrzeit,5);
END SchreibeUhrzeit;
```

Erfolgt der Aufruf immer mit dem gleichen aktuellen Parameter, leisten die Prozeduren das Gewünschte. Es zählt die Lebensdauer des aktuellen Parameters, und die Sichtbarkeit wird durch die Parameterdefinition durchgereicht. Die Lösung hat allerdings einen Nachteil: Das Hauptprogramm muß selber eine Variable einrichten, in der die Zeit abgelegt wird. Dazu müssen ein Bezeichner und ein Typ vergeben werden. Doch damit muß der Programmierer im Hauptprogramm erheblich mehr leisten, als ursprünglich gefordert war.

Sinnvoll wäre es, wenn die Darstellung der Uhrzeit und die Zählvariable nach außen nicht sichtbar wären. Damit kann sich das Hauptprogramm auf den Aufruf der Prozeduren beschränken und zudem frei Bezeichner verwenden. Wir müßten die Sichtbarkeit einer Zählvariablen Uhrzeit auf die beiden Prozeduren beschränken und gleichzeitig ihre Lebensdauer auf die des Hauptprogramms erweitern.

Dies geschieht, indem ein "lokales Modul" um die Prozeduren und die Variable gelegt wird. Ein solches Modul beschränkt die Sichtbarkeit der Bezeichner in dem Modul auf genau den Bereich des Moduls. Kein Bezeichner ist "außen" sichtbar. Zunächst der neue Programmtext - nun einschließlich eines kleinen Hauptprogramms:

```
MODULE UhrDemo;
FROM InOut    IMPORT WriteInt;

  MODULE Uhr;

  VAR Uhrzeit:INTEGER

  PROCEDURE ZaehleWeiter;
  BEGIN
    Uhrzeit:=Uhrzeit+1;
  END ZaehleWeiter;

  PROCEDURE SchreibeUhrzeit;
  BEGIN
    WriteInt(Uhrzeit,5);
  END SchreibeUhrzeit;

  END Uhr;

BEGIN
  ZaehleWeiter;
  SchreibeUhr;
END UhrDemo.
```

Ein lokales Modul erhält einen Namen und wird vom Schlüsselwort MODULE <Bezeichner> eingeleitet und mit END <Bezeichner> beendet. Alle Objekte in dem Modul sind nur darin sichtbar. Objekte von außerhalb des Moduls sind im Modul nicht sichtbar. Die Lebensdauer der Objekte innerhalb des Moduls entspricht der der Objekte außerhalb, im Beispiel wie bei globalen Objekten.

IMPORT-EXPORT-Geschäfte

Unter diesen Bedingungen kann das obige Programm nicht übersetzt werden. ZaehleWeiter ist dem Hauptprogramm nicht bekannt, daher erzeugt der erste Aufruf eine Fehlermeldung beim Übersetzen. Aber schon in SchreibeUhrzeit erhalten Sie eine Fehlermeldung, denn WriteInt ist zwar im Hauptprogramm bekannt, aber nicht innerhalb des Moduls Uhr.

Logisch, daß auf diese Weise die Module wertlos wären. Daher gibt es Mechanismen, um die Sichtbarkeit explizit festzulegen, die Im- und Exporte.

Ein Import erlaubt es, ein Objekt von außen dem Modul sichtbar zu machen. Ein Export macht ein Objekt innerhalb des Moduls nach außen sichtbar. Diese Festlegungen machen zusammen die "Schnittstelle" eines Moduls aus.

Was ist nun im Beispiel zu im- und exportieren? Das Modul Uhr soll die beiden Prozeduren bereitstellen, nicht aber die Variable. Die Prozeduren benötigen von außen die Prozedur WriteInt.

Im- und Exporte werden am Modulanfang in Listen bekanntgegeben. Die Importliste beginnt mit dem Schlüsselwort IMPORT, die Exportliste mit EXPORT. Alle Objekte werden mit Komma getrennt aufgezählt.

Das Modul sieht mit Schnittstelle also so aus:

```
MODULE Uhr;
 IMPORT WriteInt;
EXPORT ZaehleWeiter, SchreibeUhrzeit;

VAR Uhrzeit:INTEGER;

PROCEDURE ZaehleWeiter;
BEGIN
...
```

Damit erstreckt sich der Sichbarkeitsbereich von WriteInt auch auf Uhr, der von ZaehleWeiter und SchreibeUhrzeit auf das gesamte Programm.

Module können natürlich auch geschachtelt werden, genaugenommen besteht das Beispielprogramm schon aus zwei geschachtelten Modulen. Dabei kann man das eigentliche Hauptprogramm als Programm- oder Hauptmodul bezeichnen.

Initialisierungsteil

Nun ist dieses Beispiel aber immer noch nicht komplett, denn Uhrzeit wird nicht initialisiert, also auf einen bestimmten Wert voreingestellt. An irgendeiner Stelle müßte noch eine Anweisung "Uhrzeit:=0;" untergebracht werden.

Der Hinweis, daß ein Hauptprogramm ebenfalls ein Modul darstellt, zeigt schon den Weg: Jedes lokale Modul hat einen Initialisierungsteil, der wie das Hauptprogramm in einem Programmodul durchlaufen wird.

In einem Programm, das lokale Module enthält, führt der Rechner zunächst die Initialisierungsteile der lokalen Module der Reihe nach aus, um dann mit der Bearbeitung des Hauptmoduls zu beginnen. Damit lautet das vollständige Beispielprogramm:

```
MODULE UhrDemo;
FROM InOut    IMPORT WriteInt;

 MODULE Uhr;

 VAR Uhrzeit:INTEGER;

 PROCEDURE ZaehleWeiter;
 BEGIN
  Uhrzeit:=Uhrzeit+1;
 END ZaehleWeiter;

 PROCEDURE SchreibeUhrzeit;
 BEGIN
  WriteInt(Uhrzeit,5);
 END SchreibeUhrzeit;
 BEGIN
  Uhrzeit:=0;
 END Uhr;
```

```
BEGIN
  ZaehleWeiter;
  SchreibeUhr;
END UhrDemo.
```

Der Rechner arbeitet bei der Ausführung zuerst die Zuweisung an Uhrzeit ab, um dann ZaehleWeiter auszuführen. Damit liefert die Ausgabe eine "1".

Sinn und Form

Welchen Vorteil kann es nun haben, diesen vermeintlich riesigen Schreibaufwand zu treiben, wenn man, wie oben beschrieben, auch einfach eine Lösung mit Parametern und einer globalen Variablen schreiben könnte?

Die Stichworte dazu heißen Datenkapselung und Abstraktion. Durch das Verstekken der Variablen Uhrzeit in dem Modul ist sie von der Umgebung abgekapselt. Nur die dafür vorgesehenen Prozeduren können - über die Sichtbarkeitsbegrenzung - auf sie zugreifen. Damit ist sichergestellt, daß nur die dafür vorgesehenen Prozeduren Änderungen vornehmen können.

Eine "wilde" Zuweisung an Uhrzeit ist nicht möglich, und das kann nur größere Sicherheit beim Programmieren bringen. Das Problem der Uhrenbehandlung wird einmal zentral in den zwei Prozeduren gelöst, dabei können alle Eventualitäten berücksichtigt werden. So müßte z.B. noch ein Test auf "Überlauf" der Uhrzeit eingeführt werden; dies geschähe in dem Modul Uhr und nicht mehr an verschiedensten Stellen im Programm, die schwer auffindbar wären.

Zudem liefert die Beschränkung der Sichtbarkeit den Vorteil, daß in Uhr beliebige Bezeichner und Objekte benutzt werden können, ohne daß auf das Hauptprogramm Rücksicht genommen werden muß. Das Hauptprogramm seinerseits ist jederzeit beliebig erweiterbar, ohne daß man Rücksicht auf die internen Vorgänge des lokalen Moduls nehmen muß. Der Einführung einer globalen Variablen Uhrzeit steht nichts im Wege - die Sichtbarkeitsregeln sorgen dafür, daß diese nicht mit der im Modul Uhr in Konflikt kommt.

Die Abstraktion erleichtert den Umgang mit komplizierten Daten. Es ist dem Hauptprogramm völlig egal, wie Uhr die Uhrzeit weiterschaltet, wie sie dargestellt oder ausgegeben wird. Man könnte ohne weiteres die Uhrzeit in Uhr in einer Zeichenkette verwalten, am Hauptmodul ändert das nichts. Der Programmierer kann an einer Stelle Verbesserungen vornehmen, ohne daß sich damit die Bedingungen an einer anderen ändern würden.

Externe Module

Diese Überlegungen klingen fast so, als wenn ein lokales Modul am besten überhaupt nicht im selben Programmtext stehen sollte wie das Hauptmodul. Wenn sich der Programmierer mit den Funktionen eines Moduls unabhängig von den Funktionen des eigentlichen Programms befassen soll, sollte vielleicht auch der Text des Moduls unabhängig von Text des Hauptmoduls sein. Wenn die konkrete Programmierung dieser Funktionen später austauschbar sein und unabhängig vom Hauptprogramm verbessert werden soll, warum sollte dann nicht sogar der Compiliervorgang so getrennt werden, daß bei Veränderung einzelner Funktionen eines Moduls nur noch dieses Modul neu compiliert werden müßte?

Diese Forderungen sind durchaus berechtigt. Sie werden umso bedeutsamer, je größer und komplexer ein Programmierprojekt wird und je mehr Programmierer daran mitarbeiten. Auch ist es sinnvoll, für Standardprobleme eine Reihe vorgefertigter Module bereitzuhaben, die in jedem Programm verwendet werden können, ohne immer wieder den Compiler starten zu müssen. Software-Firmen könnten Funktionen verkaufen wollen, ohne den Quellcode mitzuliefern.

Dem Wunsch nach einer weitergehenden Modularisierung wird in Modula-2 mit den externen Modulen Rechnung getragen.

Externe Module teilen sich auf in ein Definitions- und ein Implementationsmodul. Im Definitionsmodul wird die Schnittstelle nach außen beschrieben, also in etwa die Exporte eines lokalen Moduls. Das Implementationsmodul enthält den eigentlichen Code für die bereitgestellten Prozeduren bzw. Objekte. Das Definitionsmodul beschreibt alles, was bei einem Import durch ein anderes Modul bekannt sein muß, also Prozedurköpfe, Typen oder "öffentliche" Variablen. Es wird durch einen speziellen Modulkopf eingeleitet: "DEFINITION MODULE".

Bleiben wir bei dem Uhrenbeispiel:

```
DEFINITION MODULE Uhr;

PROCEDURE ZaehleWeiter;

PROCEDURE SchreibeUhrzeit;

END Uhr.
```

Es werden lediglich die Prozedurköpfe notiert, die natürlich auch eventuelle Parameter enthalten könnten. Wird in einer solchen Definition ein Bezeichner verwendet, der aus einem anderen Modul stammt, muß er im Definitionsmodul importiert werden.

Völlig getrennt davon, also in einer anderen Datei, wird das Implementationsmodul - markiert durch "IMPLEMENTATION MODULE" - notiert:

```
IMPLEMENTATION MODULE Uhr;

FROM InOut    IMPORT WriteInt;
VAR Uhrzeit:INTEGER;

PROCEDURE ZaehleWeiter;
BEGIN
  Uhrzeit:=Uhrzeit+1;
END ZaehleWeiter;

PROCEDURE SchreibeUhrzeit;
BEGIN
  WriteInt(Uhrzeit,5);
END SchreibeUhrzeit;

BEGIN
  Uhrzeit:=0;
END Uhr.
```

All das, was bei lokalen Modulen nicht nach außen sichtbar war, wird hier einfach wie bei einem Hauptmodul aufgeschrieben. Es gibt einen Initialisierungsteil, der wie beschrieben vor dem des Hauptmoduls ausgeführt wird.

Es gibt allerdings die Bedingung, daß alle Dinge, die im Definitionsmodul nach außen freigegeben wurden, im Implementationsmodul ihre Entsprechung finden müssen. In einem konkreten System müssen Sie zuerst das Definitions- und dann das Implementationsmodul übersetzen (s.u.). Danach kann die Übersetzung des Hauptmoduls beginnen.

Wäre im Implementationsmodul die Prozedur SchreibeUhrzeit nicht definiert, würde der Compiler einen Konflikt zwischen der Schnittstelle und dem tatsäch-

lich vorhandenem Programmtext feststellen und das Implementationsmodul zurückweisen.

Nun existiert ein externes, vorcompiliertes Modul Uhr im Modula-System. Wie werden die Routinen in andere Hauptmodule importiert? Bei einem lokalen Modul reicht der EXPORT im Modul aus, einen Bezeichner nach außer sichtbar zu machen. Bei einem externen Modul muß sich das Hauptprogramm die entsprechenden Module und Routinen "holen", sie explizit importieren.

Am Beginn eines Hauptmoduls wird eine Importliste notiert. Darin kann z.B. stehen:

```
IMPORT Uhr;
```

Der Compiler sucht nun alle Dateien ab, bis er auf ein vorcompiliertes Definitionsmodul mit dem Namen Uhr trifft, und stellt alle darin exportierten Bezeichner bereit. Problematisch können dabei natürlich Namenskonflikte werden, wenn z.B. im Hauptprogramm dieselben Bezeichner verwendet werden. Die Modularisierung soll aber solche Bedingungen ausschließen. Die obige Zeile bewirkt einen sogenannten qualifizierten Import aller Bezeichner.

Ein so importiertes Objekt wird durch den Namen des externen Moduls und - durch einen Punkt getrennt - dem Bezeichner der eigentlichen Routine oder Variablen benannt. Das Weiterzählen der Uhr müßte in einem solchen Hauptmodul als Anweisung "Uhr.ZaehleWeiter;" notiert werden.

Falls keine Namenskonflikte auftreten können, ist eine etwas andere Importliste möglich, die die Notwendigkeit zu qualifizierten Bezeichnern aufhebt. Ein Import beginnt nun mit dem Schlüsselwort FROM, der Angabe des Modulnamens, dem Schlüsselwort IMPORT und einer Liste aller zu importierenden Bezeichner. Dabei müssen nicht alle vom externen Modul exportierten Objekte aufgezählt werden. Will ein Modul lediglich ZaehleWeiter ohne Qualifizierung verwenden, lautet die Importzeile:

```
FROM Uhr IMPORT ZaehleWeiter;
```

Mehrere importierte Bezeichner werden durch Komma getrennt:

```
FROM Uhr IMPORT ZaehleWeiter,
                SchreibeUhr;
```

Damit wissen Sie auch endlich, was sich hinter den Zeilen in den Beispielprogrammen verbirgt, etwa FROM InOut IMPORT Write, WriteString, WriteLn, WriteInt, Read;

Aus dem externen Modul InOut werden fünf Bezeichner importiert, die ohne Qualifizierung verwendet werden können. InOut ist eines der Standardmodule, die in jedem Modula-System vorhanden sein sollten.

Strings-Modul

In der letzten Folge hatte ich angesprochen, daß das Modul Strings im LPR-Modula nicht ausreichend ist. Sie finden hier in Listing 1 ein Definitions- und in Listing 2 ein Implementationsmodul, die das mitgelieferte ersetzen sollen.

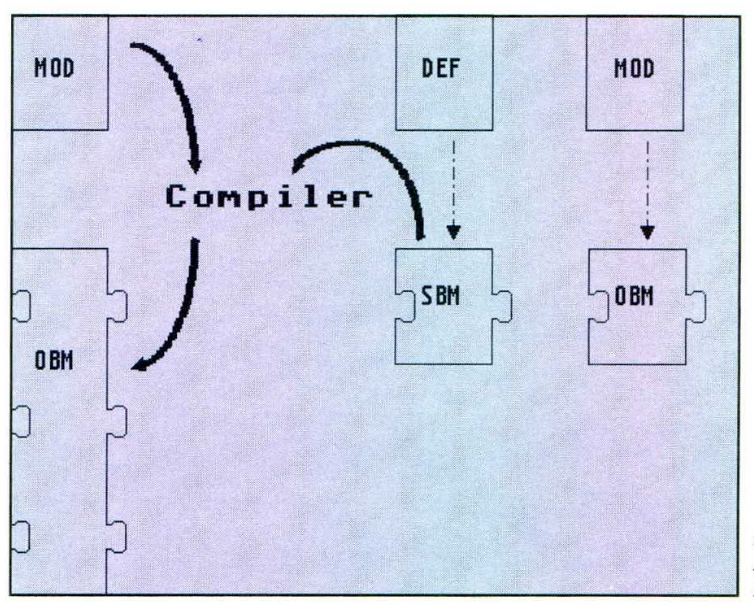


Bild 1: Generelle Arbeitsweise von Modula-Systemen.

Die Funktionen entsprechen der Definition im TDI-Modula und sollten eigentlich für die meisten Anwendungen ausreichen. Die Implementation der Routinen nimmt auf einige, wenn auch nicht auf alle, Ausnahmefälle Rücksicht.

LPR stellt String-Konstanten, die einer Prozedur übergeben werden, als Feld von CHAR dar, wobei allerdings - im Gegensatz zur Zuweisung an Variablen - kein Nullzeichen an das Ende gehängt wird. Daher muß in einigen Abfragen das Ende der Zeichenkette zusätzlich mit einem Test auf HIGH(Feld) ermittelt werden.
Es kann Fälle geben, in denen die Routinen einen Fehler erzeugen. Bitte teilen Sie solche gefundenen Fehler und falls möglich deren Lösung mit.

Die Realität

Abschließend soll noch die generelle Arbeitsweise von Modula-Systemen dargestellt werden. Aufgabe des Compilers ist es einerseits, externe Module herzustellen, und andererseits, Programmodule unter Überprüfung der Schnittstellen zu importierten Modulen zu erzeugen (siehe Bild 1).

Ein Definitionsmodul übersetzt der Compiler in eine spezielle Form, die die Schnittstellenüberprüfung ermöglicht. Typischerweise ist dies die Übersetzung einer Datei .DEF nach .SBM. Die compilierten Definitionsmodule können auch .DEF (Megamax) oder .SYM (TDI) heißen.

Das dazugehörige Implementationsmodul (.MOD) wird unter Berücksichtigung von .SBM in ein Objektmodul .OBM übersetzt. Aus diesem Grund muß immer das Definitions- vor dem Implementationsmodul übersetzt werden. Die Objektmodule können auch .MOD (Megamax) oder .LNK (TDI) heißen.

Wird nun ein Programmmodul übersetzt, nimmt der Compiler den Programmtext (.MOD oder .M), überprüft die Schnittstellen mit den importierten Modulen anhand der getrennt compilierten .SBM-Dateien und erzeugt ein Objektmodul .OBM. Darin enthalten sind Informationen über die noch zu bindenden anderen Objektmodule und den eigentlichen Code.

Im Vergleich zu C ist die Modularisierung in Modula-2 in den Compiler verlegt. Ein C-System macht per Header-File ledig-

lich die Bezeichner der Bibliotheksroutinen bekannt. Unstimmigkeiten erkennt letztlich erst der Linker. In Modula steigt schon der Compiler in die Libraries ein.

Nach dem Compilieren gibt es zwei Möglichkeiten (Bild 2). Notwendig ist auf jeden Fall ein Zusammenbinden des Hauptprogramms mit den importierten Objektdateien. Ein System mit Link-Loader hat typischerweise eine Shell, die einen Programmstart ermöglicht, bei dem der Link-Vorgang im Speicher stattfindet. Danach kann das Programm "unter der Shell" ablaufen, ohne daß eine Programmdatei erzeugt wird. Vorteile sind dabei der schnellere Link-Vorgang und die Speicherersparnis auf dem Massenspeicher und im RAM, da einige Module schon für die Shell geladen sind. Während der Programmentwicklung ist dieses Vorgehen effizient.

Um ein komplettes Programm zu erzeugen, lädt der Linker alle Objektdateien und bindet sie auf dem Massenspeicher zu einem ausführbaren Programm zusammen, das dann ohne Shell ausführbar ist. Dies ist der normale Vorgang nach dem Debugging.

In beiden Fällen sind meistens zusätzliche Objektmodule nötig, da die externen Module ihrerseits wiederum andere importieren. Hat man z.B. ein InOut-Modul, das Ausgaben in ein GEM-Fenster schreibt, ergibt sich eine wahre Link-Orgie, da der simple InOut-Import das Binden vielerlei AES- und VDI-Module nach sich zieht. Bei einigen Systemen ist es möglich, z.B. einfachere InOut-Module mit gleichen Funktionen einzubinden.

Wenn Sie ein Modula-System erwerben wollen, stehen Sie vor der fast unlösbaren Aufgabe, die "beste" mitgelieferte Sammlung externer Module auszuwählen. TDI hatte z.B. nur einfache Bibliotheken geliefert; InOut schrieb auf den "TOS-Bildschirm". Für Textfenster war der Toolkit nötig. SPC ermöglicht neben Textfenstern, mehrere Programme gleichzeitig im Speicher zu halten, mit jeweils eigener Benutzeroberfläche. Folge ist zwar ein hoher Komfort, aber auch von der Codegröße her riesige Endprodukte.

Die Modularisierung ist nicht nur ein Sprachmittel, sie bewirkt bei kommerziellen Produkten erhebliche Qualitätsunterschiede aufgrund des Lieferumfangs. Sicherlich haben Sie bei meinen Software-Tests zu Modula-Systemen in der ST-Computer bemerkt, daß ich auf diesen Gesichtspunkt meist nicht sehr detailliert eingegangen bin und nur eine allgemeine Einschätzung geliefert habe. Grund dafür ist die Tatsache, daß ein objektiver Vergleich von über 100 externen Modulen nicht mehr möglich ist. Eine Eigenart von Modula-Systemen ist es bisher, daß sie meistens aufgrund der verschiedenen externen Module wenig kompatibel sind. Es gibt einen vagen Vorschlag von Wirth über die Struktur einer Standardbibliothek. Dieser wird von den vorliegenden Systemen auf dem ST schon kaum befolgt. Eine wahre Kompatibilitätskatastrophe sind die maschinenabhängigen GEM-Module. Hier unterscheiden sich Bezeichner, Parametertypen und -reihenfolge von System zu System. Jede neue Implementierung entschuldigt sich damit, daß für Modula keine GEM-Konventionen existierten, anstelle welche zu vereinbaren.

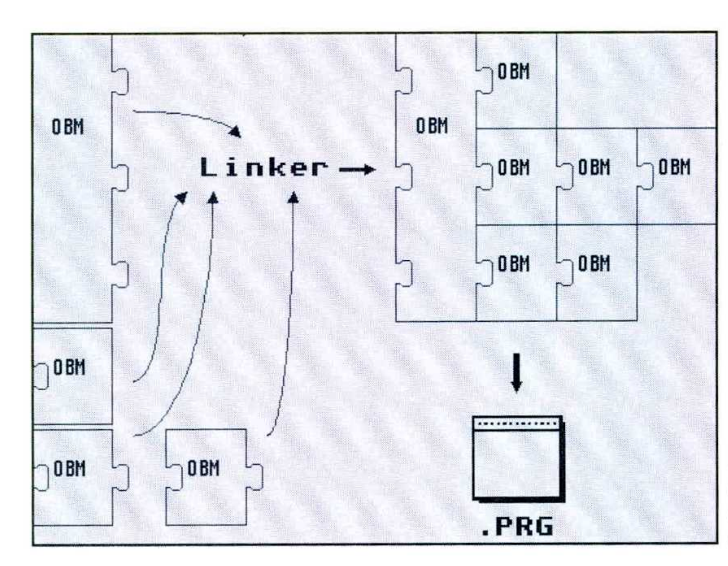


Bild 2: Zusammenbinden des Hauptprogramms mit den importierten Objektdateien.

Bezüglich der Standardmodule wird es demnächst einen DIN-Standard geben, der international abgestimmt wurde. Ich werde Sie darüber informieren, wenn ein entsprechendes, endgültiges Papier vorliegt.

RT

Hausaufgaben

1. Beschreiben Sie Sichtbarkeitsbereiche und Lebensdauer aller Variablen in folgendem - trotz fehlender Anweisungen legalem - Programm:

```
MODULE Aufgabe;
VAR a,b:INTEGER;

 MODULE m1;
 IMPORT b;
 VAR a:CARDINAL;

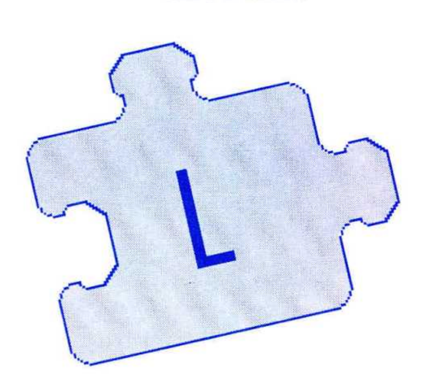
 PROCEDURE p;
 VAR b:INTEGER;
 END p;

 END m1;

PROCEDURE p;
VAR b:INTEGER;
END p;

END Aufgabe.
```

2. Schreiben Sie ein externes Modul, in dem eine Ampel simuliert wird. Nach außen sichtbar soll ein Typ zur Darstellung von Ampelfarben und zwei Funktionen SchalteWeiter und SchreibeAmpel sein. Programmieren Sie ähnlich den Beispielen in der letzten Folge mit SETs. Die Ampel soll auf Grün initialisiert werden.



dBMAN

TEL: 089/4480691
FAX: 089/4483820

by COMPUTER MAI

CAS Computer-Art-Studio

Karl Heinz Klein - Schützenstraße 12
6839 Oberhausen-Rheinhausen 1
☎ 07254-3698

Neu! Neu!

Für ATARI ST und CALAMUS!

CALAMUS-Rastergrafiken in feinster Qualität! (400 dpi). **Große Grafikbibliothek!**

Scannen und digitalisieren Ihrer Vorlagen aller Art bis DIN A4 und Umwandlung in Ihr ATARI ST-Grafikformat (Degas, Stad, GFA, Calamus). (400 dpi!)

Herstellung von individuellen Drucksachen, wie Glückwunsch-,Visiten-,Speise-und Tischkarten, Einladungen, Briefbogen, Urkunden uvm. (max. DIN A4)

Ausdruck mit Laserdrucker (s/w) oder im **Prägedruck** in vielen schillernden Farben (auch gold und silber) auf beliebigem Papier oder Karton bis DIN A4.

Erstellung von reprofähigen Layouts für Drucksachen aller Art bis DIN A4, zB. Vereinszeitungen, Handzetteln, Anzeigen, Werbeprospekten usw.

Günstige Preise auch bei kleineren Stückzahlen!

Unbedingt kostenlosen Grafik-Katalog anfordern!!

Unübersichtlich
wird langsam das PD-Angebot für den Atari ST.

Wenn Sie also keine Lust haben, sich stundenlang durch riesige, nichtssagende Listen zu kämpfen, dann sind Sie bei uns genau richtig!

Unsere PD-Sets enthalten auf je 10 doppelseitigen Qualitätsdisketten jeweils 7 MB, aus 700 Disks ausgewählte, Top-Software. Hier hat der Durchschnitt keine Chance.

PD-Set A (von allem etwas)
PD-Set B (Spiele für S/W)
PD-Set C (Spiele für Farbe)
PD-Set D (Anwendungen)
PD-Set E (Utilities)
PD-Set F (Grafik + Pics)
PD-Set G (Musik, div. Demos)

je **39,-** DM

Sets B – G zusammen (>40 MB) 199,-
(= 35 DM gespart)

Alle Preise incl. Disks, MwSt, Versand- und Verpackungskosten sowie unserer Katalogdiskette. Wir liefern innerhalb von 24h – garantiert.

Gerald Köhler PD-Schnellversand Atari ST
Mühlgasse 6 · 6991 Igersheim
Hotline: 07931/43922 + 44661



Chemo - Soft

*Softwareentwicklung * DTP - Service*
EDV - Versandhandel

Nadorster Str. 81 2900 Oldenburg
Telefon: (0441) 82851
Geschäftszeit: Montag bis Freitag 13⁰⁰- 19⁰⁰ oder nach Vereinbarung

SIGNUM!ZWEI	348,-	LDW POWER CALC	210,-
IMAGIC	390,-	SUPERBASE 2	168,-
CREATOR	210,-	ANTI-VIREN-KIT II	85,-
Mega Paint 2.x	330,-	GFA 3.0 [Int. + Compiler]	165,-
D.R.A.W. 3.0	115,-	ST-MATH	85,-
Sidepaint	72,-	ST-CHEMPLOT	85,-
BECKERcalc	440,-	Picture Disk 1000	85,-
BECKERcad	440,-	GFA DRAFT plus	290,-
CALAMUS junior	345,-	CYBER CAD-3D 2.0	130,-
DMC´s "Outline"	325,-	PEGASUS ST3	299,-
FIRST WORD PLUS 3.11	215,-	fibuMAN e	345,-
THAT´S WRITE	240,-		
TEMPUS 2.x	108,-	VORTEX HDplus 20 MB	979,-
Starwriter	170,-	VORTEX HDplus 30 MB	1128,-
Omikron COMPILER 3.0	159,-	VORTEX HDplus 60 MB	1749,-
Omikron Assembler	110,-	WECHSELPLATTE 44MB	2598,-
Omikron EasyGEM	72,-	STREAMER 150MB	2998,-
ADIMENS V2.3	210,-	ALADIN + ROM´s	550,-
ADITALK V2.3	210,-	3,5" Floppy anschlußfert.	270,-

CHEMIEPROGRAMME:

CHEMOTECH V1.0 — **159,-**
CHEMOTECH Spezialversion — **199,-**
[Korrosionsgrößen / Abtragungsraten / Umrechnungen / Oberflächenberechn.]
KRISTALLOTECH V1.0 [Kristallstrukturen] — **89,-**

Fordern Sie doch unsere Preisliste bzw. Info-Material an!

COMPUTER-ZUBEHÖR HERGES

Ober Rischbachstr. 88 · 6670 St. Ingbert · Tel. (0 68 94) 38 31 78 · Telefax (0 68 94) 38 28 55
Geschäftszeiten Mo – Fr 9⁰⁰ – 12⁰⁰ und von 14⁰⁰ – 17³⁰ Sa von 8⁰⁰ – 12⁰⁰

Atari-Produkte:

Mega-ST1 + SM-124 komplett	1728,-
Mega-ST2 + SM-124 komplett	2328,-
Mega-ST4 + SM-124 komplett	3388,-
Aufpreis MegaST + SC-1224	313,-
Atari-Laserdrucker SLM-804	2800,-
Mega-ST1 ohne Monitor	1428,-
Mega-ST2 ohne Monitor	2028,-
Mega-ST4 ohne Monitor	3088,-
Aufpreis Mega-ST + 2. Laufw. 3.5 int.	300,-
Aufpreis Mega-ST + SLM-804	2722,-
Monitor SC-1224	688,-
Monitor SM-124	388,-
Megafile 30 komplett	948,-
Megafile 60 komplett	1728,-
Aufpr. MF-XX + Floppy 3.5 int.	333,-
Megafile 44 (44-MB Wechsel)	2222,-
Orig. Atari-Megatastatur als Zweit-, Ersatz-, Umbautastatur	348,-
1st-Word Plus/1st-Mail kompl.	148,-
Calamus „Desktop Publishing"	a. A.

ST-Floppy's + Rohlaufwerke:

Teac 3.5-T1 726 KB	238,-
Teac 5.25-T1 726 KB + 40/80 Tr.	288,-
Teac 5.25-T2 wie oben, plus: Led's für Ein/Aus + 40/80 Tr. Kein Headloadgeräusch mehr. Motor läuft nur bei Select	338,-
Teac FD-55-FR/GFR + ST-Modif.	228,-
Teac FD-135/235-FN/ – HFN	188,-

Drucker + Zubehör:

LC-10, LC-10 Color, LC-24/10, NX-1000 zu super Preisen	a. A.
Drucker von NEC und Epson	a. A.
Farbband LC-10	8,80
Farbband LC-10 Color	20,-
Farbband LC-24/10	15,-
Druckerkabel Atari-Centr.	15,-
Autom. Blatteinzug LC-10/-24/10	188,-
Druckerumschalter aller Art	ab 66,-

Gehäuse für:

Teac FD135/235 oder NEC 1037a	14,-
Teac FD-55-FR, -GFR oder comp.	22,-
3.5 Gehäuse + Netzteil aufgeb.	46,-
5.25 Gehäuse + Netzteil aufgeb.	55,-
Universal-Kleingehäuse	a. A.

Netzteile für Eigenbau:

Schaltnetzteilplatine +5V,5A./+12V,2.5A./–12V,0.5A. Platine kpl. 160mm ★ 100mm ★ 45mm	88,-
Steckernetzt. 1amp. 1,5 – 12 Volt	19,-

Sonstiges für ST:

GFA-Basic V.2.0	22,-
Dataphon s21 d-2 + ST-Kabel	a. A.
Buskabel 3.5 oder 5.25 kpl.	22,-
Blitterchip für ST	148,-
Floppy Umschalter 3-fach	49,-
Floppy-/Monitorst./-Buchse	je 4,50
Atari-Monitorumschalter	39,-
GFA-Basic + GFA-Compiler V2.0	49,-

Wir liefern alle Atari-Computer, alle Atari-Ersatzteile, Halbleiter, GFA-Produkte, IC's + Sockel, Lüfter, Disketten uvm: **a.A.**

Aktuelle Preise – auf Anfrage – bitte per Telefon erfragen!
Lieferung per UPS zzgl. Porto + Verpackung. Alle Angebote freibleibend!
Selbstabholung Mo – Fr v. 14⁰⁰ – 19⁰⁰ & Sa v. 12⁰⁰ – 14⁰⁰ nur gegen Termin

---

AM BRAHMKAMP 8 · 2800 BREMEN 33
TELEFON 04 21 / 23 42 64 · FAX 23 95 84

FOTOSATZ
QUALITÄT mit Calamus in 1270 dpi

Wir belichten Ihre Calamus-Dokumente mit Linotype-Laserbelichter 300.
Diese Anzeige wurde mit Calamus/Atari erstellt und in 1270 dpi auf Fotosatz-Film belichtet.
Informieren Sie sich bei uns!



GRAFIK+REPROTECHNIK

ATARI ST SOFTWARE

Sie suchen ein leistungsfähiges Programm, mit dem Sie Ihre gesamte Auftragsabwicklung durchführen können???
Dann sollten Sie

ST-AUFTRAG

unbedingt kennenlernen!!!

ST-AUFTRAG ist ein Programm zur Auftragsbearbeitung (Angebote, Auftragsbestätigungen, Lieferscheine, Rechnungen, Bestellungen, Gutschriften) mit integrierter Artikel- und Adressenverwaltung. Der Dialog zwischen dem Anwender und ST-Auftrag wird über die komfortable Benutzeroberfläche GEM geführt. Zusätzlich sind alle wichtigen Funktionen über Tastaturkommandos erreichbar. ST-Auftrag enthält weder Kopierschutz, Hardwareschlüssel noch sonstige Anwenderhindernisse. Die Installation des Systems auf Festplatte in jedem beliebigen Ordner ist problemlos möglich.

Leistungsbeschreibung in Stichworten:

Max. Anzahl von Artikeln und Adressen: 65535; Artikellisten und -etiketten; Adressenlisten, -etiketten und Serienbrieferstellung; Offene-Posten-Liste; Bestellvorschlagsliste; autom. Mahnwesen; Lagerüberwachung; Rabattliste für jeden Artikel separat; alle Ausgaben auf Drucker, Bildschirm oder Datei; Ausgaben mit 1st Word Plus Formularen (d.h. alle Schriftattribute, Schriftbreiten, Kopf- und Fußzeilen werden mit ausgegeben); Originalgetreue Ausgabe auf Bildschirm; Artikel mit Langtext (ab Version 1.10); Barverkauf; 10 frei eingebbare Liefer/Versandarten; Ausfüllen von Nachnahmezahlkarten...

Hardwarevoraussetzungen:

Monochrommonitor, doppelseitiges Diskettenlaufwerk, Festplatte empfehlenswert, ATARI ST oder MEGA ST mit mindestens 1 MB RAM.

Der Preis: nur DM 398.-

Und damit Sie ST-AUFTRAG ausgiebig testen können, gibt es eine Demoversion, die keinerlei Funktionsbeschränkungen hat (max. Anzahl Adressen 2, Artikel 5). Die Demoversion kostet DM 20.- (mit Handbuch DM 50.-). Die Preise für Demoversion und Handbuch werden beim Kauf voll angerechnet!

Versand per Vorauskasse oder per Nachnahme (plus DM 5.-).

AS DATENTECHNIK
AS-DATENTECHNIK, MAINZER STR.69, D-6096 RAUNHEIM, Tel.: 06142 / 2 26 77

ATARI ST SOFTWARE

AS-HAUSHALT (Haushaltsbuchführung) — DM 99.-
Buchführung für alle Privathaushalte. Buchen über Konto/Gegenkonto. Suchen, Löschen und Ändern von Buchungen. Auto-Save. Monats- und Jahresabschluß. Universelle Druckeranpassung. Journalausdruck. Statistik als Balkengrafik. Deutsches Handbuch.

ST-FIRMENBUCH (Buchführung für Gewerbetreibende) — DM 139.-
Doppelte Buchführung für Gewerbetreibende. Automatisches Mitführen der MwSt.-Konten. Fünf versch. MwSt-Sätze frei einstellbar. Universelle Druckeranpassung. Abschlußzeitraum Monat, Quartal oder Jahr. Einnahmen/Überschußrechnung. Ausgabe eines Journals. Umsatzsteuervoranmeldung. Ausführliches deutsches Handbuch.

ST-ÜBERWEISUNGSDRUCK — DM 45.-
Bedruckt alle Arten von Überweisungsträgern, Schecks, Zahlkarten usw. Universelle Druckeranpassung, eigene Formulare können mit einem Texteditor selbst erstellt bzw. angepaßt werden. Umwandlung des Betrages in ein Zahlwort. Die Eingaben sind speicherbar.

DATENBANKANWENDUNGEN — DM 69.-
6 Anwendungen für Adimens ST, z.B. Termine/Adressen, Lager, Videocassetten, Schülernoten, Zeitschriftenartikel, Bibliothek.

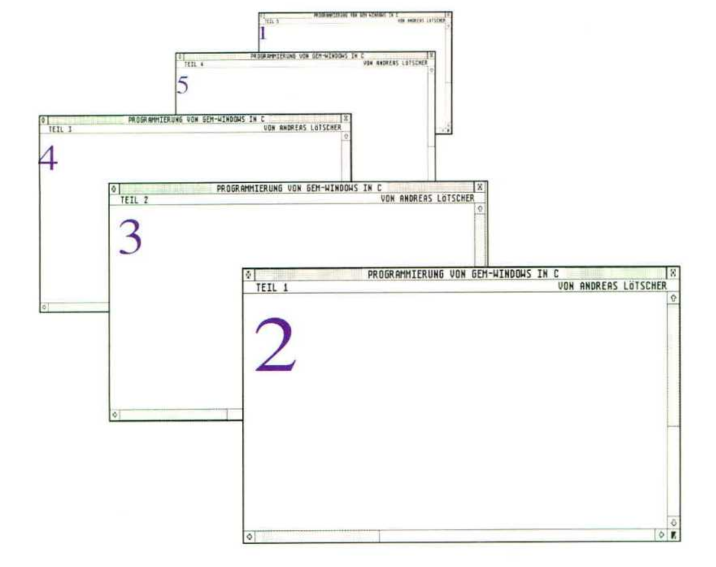
ST-SCHREIBMASCHINE — DM 59.-
Mit diesem Programm arbeitet Ihr Atari ST + Matrixdrucker wie eine Schreibmaschine. Sehr gut geeignet zum Ausfüllen von amtlichen Formularen. Ausdruck entweder direkt oder über editierbares Display. 10 Floskeltasten definierbar. Druckeranpassung. Deutsche Anleitung.

ST-ETIKETT Bedruckt alle Arten von Etiketten. Ausgabebild selbst anpaßbar. Mit Seriennummerngenerierung. — DM 59.-

***NEU:* ST-TRAINER MATHEMATIK** — DM 79.-
Lern- und Trainingsprogramm für Schüler (1. bis 5. Schuljahr). Abfrage der vier Grundrechenarten, Bruchrechnen, Kürzen, Erweitern, Längen-, Flächen-, Raum- und Gewichtsmaße (Umrechnungen). Die Abfrage- und Benotungsbereiche sind frei einstellbar. Protokolldruck optional. GEM. Mit deutschem Handbuch.

INFO 1/89 kostenlos!
Lieferung per Vorauskasse (V-Scheck) oder per Nachnahme (plus DM 5.- Versandkosten)!

AS DATENTECHNIK
AS-DATENTECHNIK, MAINZER STR.69, D-6096 RAUNHEIM, Tel.: 06142 / 2 26 77



WINDOWS UNTER GEM

Das "Snappen" von Fenstern

In dieser Folge der Programmierung von Fenstern mit allem Drum und Dran unter GEM soll es um das richtige "Snappen" der Fensterkoordinaten gehen. Was das bedeutet und wie es zu realisieren ist, erfahren Sie weiter unten. (Ganz Neugierige werfen schon mal einen Blick in Listing 1).

Bevor wir aber mit dem neuen Stoff beginnen, verpacken wir die in der letzten Ausgabe entwickelten Programmteile in handliche Funktionen, um mit diesen unsere Windowbibliothek anzurichten. Dazu verpacken wir den Programmteil innerhalb der *do{..}while-Schleife* (vgl. Listing 2 in der letzten Folge) in eine Funktion namens *handle_window()*, der wir den von *evnt_mesag()* gelieferten Message-Buffer übergeben. Das Ergebnis sehen Sie in Listing 1, das übrigens den aktuellen Zwischenstand unserer gesamten bisherigen Window-Bibliothek darstellt.

Änderungen sind vorbehalten ...

Den neuen Funktionen (und späteren Erweiterungen) zugunsten habe ich einige Neuerungen eingeführt, wobei auch ältere Funktionen angepaßt werden mußten. So können Sie unschwer feststellen, daß *open_window()* einige Änderungen über sich ergehen lassen musste. Die schwerwiegendste Neuerung betrifft die Einführung einer globalen Variablen, die alle wichtigen anfallenden Daten über unsere Fenster aufnimmt und sie bei Bedarf zur Verfügung stellt. Dazu dient uns in C eine Struktur wie diese:

```
struct wind_data
{
    char  name[80];
    GRECT max;
    WORD  elements;
    WORD  align;
    WORD  snap;
    WORD  full;
};
```

Damit verknüpfen wir Variablen unterschiedlichen Typs zu einer neuen Einheit namens *wind_data*, um auf sie als Ganzes zugreifen zu können. Dann definieren wir mit

```
struct wind_data  windows[8];
```

ein Feld von acht Elementen des Typs *wind_data*. D.h. uns wird für acht Fenster Speicher reserviert, damit wir für jedes den Namen (name), die Maximalgröße (max), die Bestandteile (elements), den Faktor zur horizontalen Ausrichtung (align), ob wir das Fenster snappen wollen (snap) und ob es auf der maximalen Größe ist (full), speichern können. (Was align und snap genauer bedeuten, siehe weiter unten!)

Um auf ein Element unserer Struktur zuzugreifen, verwenden wir den Punktoperator; nun kann z.B. *wind_open()* mit der Anweisung "windows[w_handle]. max.g_x = mx;" die größte X-Koordinate des Fensters mit der Identifikation *w_handle* unserer Struktur zuweisen (vgl. Listing 1).

Ebenso wurde die Funktion *handle_full()* angepaßt. In der vorliegenden Version speichert sie die Information, ob das Fenster gerade auf der maximalsten Größe ist oder nicht, mit windows[w_hndl] .full = TRUE; (bzw. FALSE) ab.

Aber warum?
Die Antwort auf diese Frage erfahren Sie im folgenden:

Was heißt eigentlich "snappen"?

Ein Programm sieht es vielleicht nicht gerne, wenn der Benutzer ein Fenster teilweise aus dem Bildschirm schiebt. Um dies zu verhindern, können wir das ganze Fenster snappen, d.h. es in den Bildschirm zurückschieben. "Snappen" bedeutet also, dafür zu sorgen, daß ein Fenster immer ganz innerhalb eines bestimmten Bereiches zu sehen ist (meist auf dem ganzen Desktop, natürlich ohne Menüleiste!).

Hinzu kommt, daß man dafür sorgt, daß die Koordinaten (der Arbeitsfläche) unseres Fensters nur ganzzahlige Vielfache eines definierten Wertes annehmen. Dies ist z.B. nützlich, wenn man in ein Fenster Text ausgeben möchte. Denn sind die Fenstergrenzen an Wortgrenzen ausgerichtet (Snapwert 8), läuft die Textausgabe etwas schneller ab. Für diese beiden Vorhaben brauchen wir zwei Funktionen, deren Arbeitsweise ich im folgenden vorstellen werde.

Zum ersten...

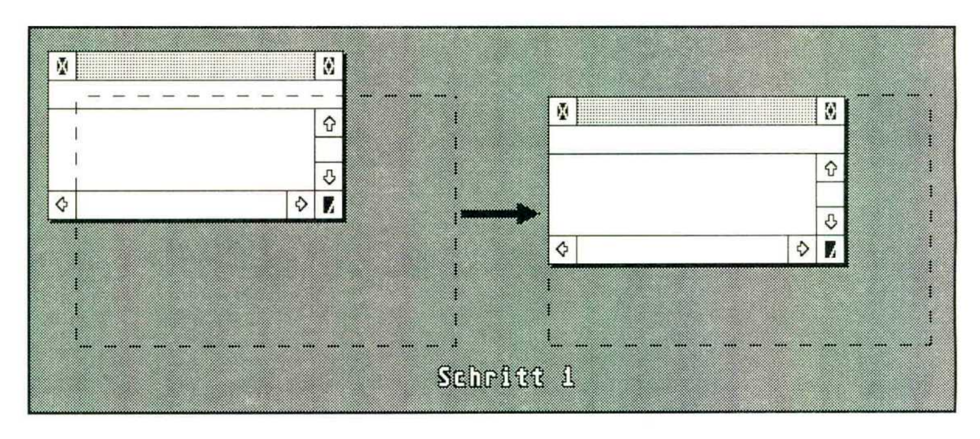
snap(): erwartet als Parameter zwei Pointer auf eine GRECT-Struktur, d.h. zwei Pointer auf die Koordinaten eines Rechtecks. Das erste definiert den Bereich, in dem das Fenster sich befinden soll [bei Aufruf normalerweise gleich die *open_wind()* mitgeteilten Maximalkoordinaten], und das zweite spezifiziert die Größe des Fensters.

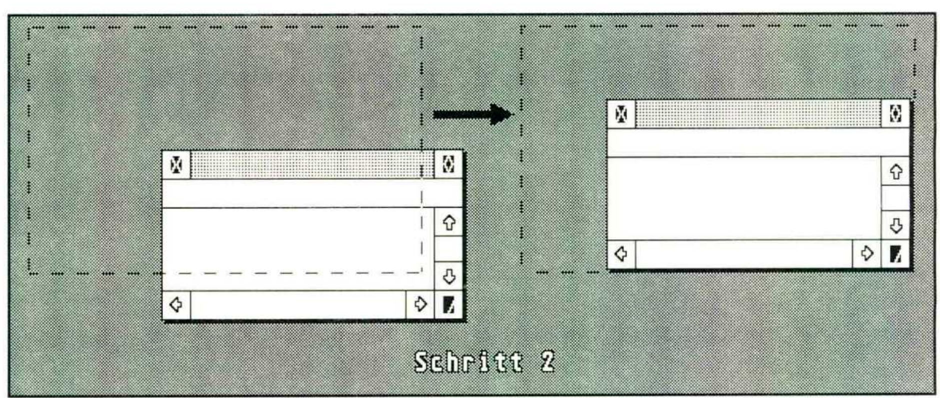
In einem ersten Schritt vergleicht die Routine die X- und die Y-Koordinaten des Fensters mit denen des definierten Bereichs. Falls das Fenster außerhalb des Bereichs liegt (d.h. falls die X- bzw. die Y-Koordinate kleiner als die des Bereichs ist), schiebt *snap()* das Fenster in den Bereich zurück. Bild 1 veranschaulicht die Problematik hierzu.

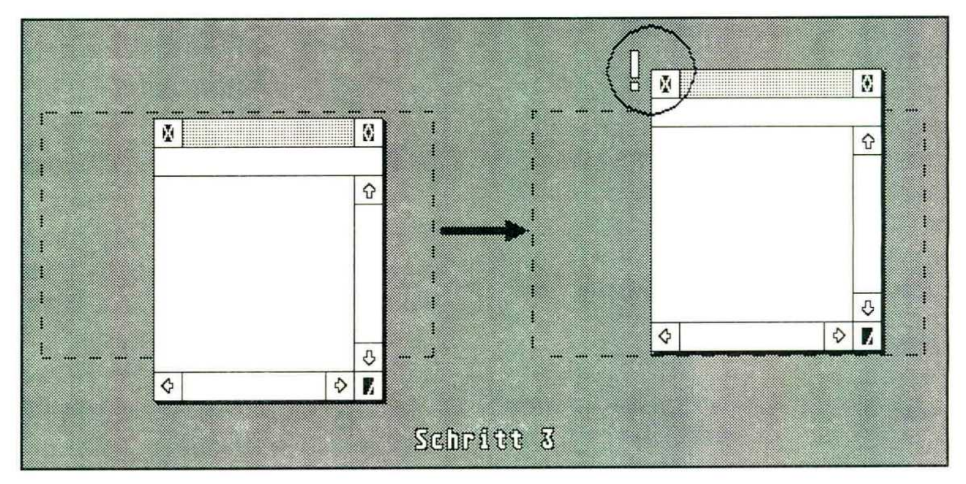
Im zweiten Schritt überprüft *snap()* die rechte untere Ecke des Fensters. Wenn deren Koordinaten (Addition von X-Koordinate mit der Breite bzw. von Y-Koordinate mit der Höhe) außerhalb des Bereichs liegen, verfährt *snap()* mit dem Fenster wie in Schritt 1. Nun sollte unser Fenster eigentlich vollständig innerhalb des Bereiches liegen, Bild 3 zeigt aber, daß das nicht immer der Fall sein muß. Angenommen, das Fenster besitzt schon fast seine maximale Höhe und der Benutzer betätigt den SIZER-BUTTON, um es noch weiter zu vergrößern, dann passiert folgendes: Schritt 1 tritt nicht in Kraft, da die Größenänderung keinen Einfluß auf die X- bzw. Y-Koordinaten des Fensters hat. In Schritt 2 aber setzt *snap()* die rechte untere Ecke des Fensters gleich der des Bereiches, da die des Fensters infolge der Größenänderung außerhalb des definierten Rechtecks zu liegen kam. Jetzt befindet sich die Y-Koordinate des Fensters aber außerhalb des Bereiches (vgl. Bild 3)! Es wird also nötig, einen dritten Schritt einzuführen, der, falls der oben geschilderte Zustand eintritt, die X-Koordinate und die Breite bzw. die Y-Koordinate und die Höhe gleich den Werten des einschränkenden Rechtecks setzt. Damit liegt unser Fenster immer innerhalb des angegebenen Bereiches!

Es drängt sich auf, *snap()* innerhalb von *handle_window()* bei Auftreten einer *WM_SIZED* - bzw. einer *WM_MOVED*-Mitteilung zu benutzen, um dem Anwender auf die Finger (bzw. auf die Maus) zu schauen (vergleiche Listing 1):

```
if(windows[buffer[3]].snap)
snap(&windows[buffer[3]].max,&buffer[4]);
```







Snappen eines Fensters

Mit Hilfe der if-Anweisung wird kontrolliert, ob der Programmierer das Snappen bei diesem Fenster wünscht [vgl. Funktionsargumente bei *open_window()*]. Falls ja, dann führen wir dies aus und übergeben *snap()* als Begrenzung die in *windows[buffer[3]]*.max enthaltenen Maximalkoordinaten und als zu snappende Werte die von *evnt_mesag()* in *buffer[4]* bis *buffer[7]* übergebenen neuen Koordinaten.

[**Anmerkung:** Eigentlich erwartet *snap()* Pointer auf GRECT-Strukturen. Im ersten Fall ist dies zwar erfüllt (der Operator "&" liefert die Adresse einer Variablen), doch im zweiten Fall übergebe ich die Adresse von *buffer[4]*, was zwar nicht ganz sauber ist. C aber durchläßt, denn es funktioniert, da ich ein Array von n WORD-Variablen (WORD array[n];) auch als Struktur von n WORD-Variablen (struct Array{ WORD x1,x2,x3,...,xn}array;) beschreiben kann, und die einzelnen Variablen im Speicher bei beiden Varianten hintereinander gespeichert werden. (Der Zugriff auf die einzelnen Elemente ist natürlich verschieden, array[n] bzw. array.xn).].

... und zum zweiten

Die Funktion *align()* ermöglicht es uns, einen Koordinatenwert auf ein ganzzahliges Vielfaches einer Konstanten zu begrenzen. (Ich führe das Snappen nur für die X-Koordinate durch, falls Sie auch die anderen Koordinaten snappen möchten, können Sie mit geringem Aufwand die Bibliothek dementsprechend abändern !)

Wenn es um Beratung, Service und gute Preise für Sie geht :

z. B. Software

PHONE — **DM 78,-**

Die bequeme Adressverwaltung, um ständig auf dem laufenden zu sein

- als Accessory jederzeit aufrufbar
- schnell durch ausgereifte Programmiertechnik
- Daten aus anderen Programmen durch Funktionstasten abrufbar

ST-DCL — **DM 298,-**

Kommandozeileninterpreter + Tools

ST-DCL erlaubt es, auf einem Atari unter einer Kommandoumgebung zu arbeiten, die der VAX /VMS-Digital Command Language (DCL) nachempfunden ist. Auch für MS-DOS lieferbar.

PRIVATLIQUIDATION — **DM 375,-**

PRAXISBUCHHALTUNG — **DM 375,-**

Arzt-Software

Aus der Praxis für die Praxis. Praxisliquidation und Praxisbuchhaltung. Info anfordern. Demo plus Handbuch für DM 40,-. (Wird bei Kauf angerechnet.)

Sämtliche PD- u. Standard-Software lieferbar: Application Systems, C.A.$.H., CCD, DMC, Drews EDV+BTX, G Data, GFA, HAGERA, Hyper-Soft, Kniss-Soft, Kieckbusch, NovoPlan, Technobox u.v.m.

z. B. Hardware

Atari 1040 STF (komplett)	**DM**	**1.498,-**
Atari Mega ST 2 (inkl. GFA-Basic)	**DM**	**2.698,-**
vortex HD plus 20	**DM**	**998,-**
vortex HD plus 30	**DM**	**1.198,-**
vortex HD plus 60	**DM**	**1.798,-**
star LC 10	**DM**	**598,-**
NEC P6 plus	**DM**	**1.698,-**

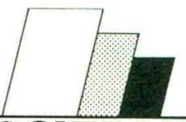
Alle Drucker mit deutschem Handbuch und Kabel. Sämtliche Hardware für Atari-Computer lieferbar. Alle Angebote freibleibend.

MAUSO ST — **DM 128,-**

Die Mäuse sind los:

RS 232 Maus für Atari lieferbar. Die alternative Maus. Inklusive Software.

Hardware
Software
Service



LOGITEAM

Computerhandelsgesellschaft mbH
Kölner Str. 132 - 5210 Troisdorf
Telefon (02241) 7 18 97 / 98

KaroSoft

ST - Soft- u. Hardware Vertrieb

Anwenderprogramme:

ADIMENS/ADITALK Vers. 2.3	je 228,–
Retouche, digitale Bildverarbeitung	385,–
Tempus 2.0	119,–
Desk Assist/4 +	189,–
Steve Vers. 3.1	478,–
Scarabus, Signum II – Fonteditor	95,–
ST-Archivar	89,–
ST-Learn Vers. 3.5	69,–
Turbo ST	79,–
Interlink ST	79,–
Revolver	129,–
CopyStar 3.0	159,–
Timeworks DTP (GST)	nur noch 239,–
Signum II, Text/Grafikprogr.	auf Anfrage
alle Fontdisk. f. Signum lieferbar	
Headline zu Signum	89,–
STAD, Vers. 1.3	169,–
Flexdisk 1.2	66,–
Disk Utility	66,–
Protos	66,–
Daily Mail	175,–
Megamax Laser C	348,–
Creator (Appl. Systems)	229,–
Soundmachine ST	148,–
MegaPaint II, V. 2.12	378,-
OMIKRON BASIC-Compiler	169,–
OMIKRON Turbo-Assembler	99,–
BS – Handel	498,–
BS – Fibu	548,–
ST Pascal plus, Vers. 2.06	228,–
STAR-WRITER-ST, Aktionspreis:	139,–
LDW – Powercalc	245,–
PC-ditto, Eurovers. 3.96	198,–
G Copy II	95,–
Anti-Viren-Kit II	95,–
fibuMAN e/f/m, 3.0,	368,– /738,– /938,–
TKC-Haushalt-Expert	129,-
Spectrum 512	149,–
Midisoft-Studio, Mehrspursequenzer	149,–
Cyber Paint 2.0	129,–
Cyber-Studio CAD 3D 2.0	179,–
1st Proportional Vers. 3.0	115,–
Exercise/Exercise plus	79,– /99,–
Steinberg „twelve“ 12-Spur-Sequ.	99,–
BasiCalc	78,–
Querdruck	58,–

Spiele:

Popolous	65,–
F 16 Combat Pilot, dt. Handbuch	67,50
Battlehawks 1942 (Lucasfilm)	59,–
Dungeon Master, kpl. dt.	69,–
Kampf um die Krone, kpl. deutsch	69,–
Flight Sim. II, deutsche Version	99,–
Scenery Disks: 7/9/11/Japan/Eur.	je 49,–
Dakar 89	55,–
Carrier Command, dt. Handbuch	69,–
Starglider II, dt. Handbuch	69,–
Powerdrome, dt. Handbuch	69,–
Microprose Soccer	66,–
STOS, The Game Creator	79,–
STOS, Compiler	49,–
STOS, Sprites 600	39,–
STOS, Meastro (Musikprogr.)	62,–
Police Quest II	69,–
F.O.F.T.	84,50
Kings Quest IV	79,50
Tom & Jerry	55,–
ELITE, dt. Handbuch	69,–
Lords of Conquest, deutsche Vers.	55,–
Kennedy Approach	74,50
Winter Edition	49,–
Times of Lore, kpl. deutsch	75,–
Wall Street Wizard, kpl. deutsch	65,–
Kaiser, kpl. deutsch	119,–
Lombard RAC Ralley, deutsche Vers.	69,–
F 16 Falcon, deutsches Handbuch	74,50
Operation Neptun, deutsch	65,–
Zak McKracken, kpl. deutsch	69,–

Hardware:

A-MAGIC-Turbodizer mit neuer starker Software Vers. 2.0	358,–
Softwareupdate, Turbodizer 2.0	49,–
Mausmatte, dt. Qualitätsware	14,50
Hochwertiges 3,5'' Laufwerk 1 MB	289,–
AS Soundsampler Maxi m. Software	289,–
AS Soundsampler III, 16 Bit	588,–
Hardwareuhr, Einbau ohne löten	79,–
Handy Scanner IV 400 dpi, m. Softw.	798,–
Farbbänder für diverse Drucker	
Eickmann Festplatten, alle Versionen auf Anfrage	
Joystick „KONIX NAVIGATOR“	48,–

UPS-Express: Vorkasse DM 4,– Nachnahme DM 8,–

Rufen Sie uns an oder schreiben Sie uns:

Jürgen Vieth
Biesenstraße 75
4010 Hilden
Telefon 0 21 03 / 4 20 22

Katalog kostenlos



Der Testsieger.*

Für problemlose Einnahme-Überschuß-Rechnung und Finanzbuchhaltung



*Vergleichstest ST Magazin in 10/88

Fazit: „Die beste Finanzbuchhaltung für den ST.“

fibuMAN-Programm ab 398.- DM

Demo wird beim Kauf angerechnet 65.-

Zusatzmodule auf Anfrage

Schicken Sie mir
- O Demo
- O Info uvb.
- O MS-DOS
- O Atari-ST

Name ____________
Firma ____________
Str., Nr. ____________
Plz, Ort ____________
Telefon ____________



HARDTSTR. 21 4784 RÜTHEN 3
TELEFON 0 29 52 / 80 80
01 61 - 221 57 91
FAX: 0 29 52 - 32 36

Zuerst erhöhen wir die Koordinate um die Hälfte des Align-Wertes minus eins (n>>1 bedeutet, daß die Variable n um 2^1 nach rechts geshiftet (geschoben) wird, was einer Division durch zwei entspricht). Dann teilen wir diesen Wert durch den Align-Wert, nehmen davon den ganzzahligen Teil und multiplizieren ihn wieder mit dem Align-Wert, womit wir bereits das ausgerichtete Ergebnis hätten. Beispiel: k=111 und v=16. Zuerst erhöhen wir k um 16 geteilt durch 2 minus 1, also 7: k wird 118. Dann teilen wir dies durch 16 und nehmen den ganzzahligen Teil: ergibt 7. Dies mal 16 ergibt den gesuchten Wert 112, der ein ganzzahliges Vielfaches von 16 ist und dem ursprünglichen Wert 111 am nächsten liegt.

Und wie geht's weiter?

In der nächsten Folge werde ich Ihnen endlich auch die Techniken zum "Redrawing" von Fenstern präsentieren. D.h. Sie werden in der Lage sein, mehrere Fenster gleichzeitig und überlappend auf dem Bildschirm darstellen zu können. Außerdem werden wir in die Fenster auch noch etwas ausgeben - denn bisher waren sie ja nur weiß und ohne Inhalt...

Andreas Lötscher

```
  1:        /* Window-Bibliothek                         */
  2:        /* Andreas Loetscher 1988                    */
  3:      /* verwendeter Compiler : Lattice C 3.04 */
  4:     /*****************************************/
  5:
  6:  #include <a:\headers\portab.h>
  7:  #include <c:\listing1.h>
  8:
  9:  WORD    ap_id, handle, work_in[12], work_out[57];
 10:
 11:  struct wind_data   /* Struktur,die Wissenswertes */
 12:  {                  /* über unsere Fenster enthält */
 13:    char  name[80]; /* Fenstername */
 14:    GRECT max;      /* Maximalgröße */
 15:    WORD  elements; /* Bestandteile des Fensters */
 16:    WORD  align;    /* Faktor z.hor. Ausrichtung */
 17:    WORD  snap;     /* Fenster snappen(TRUE/FALSE) */
 18:    WORD  full;     /* Full-Flag (TRUE/FALSE) */
 19:  };
 20:
 21:  struct wind_data  windows[8];
 22:
 23:
 24:  void gem_init()    /* erledigt die leidigen
 25:                        Gem-Initialisierungen
 26:                        am Anfang */
 27:  {       WORD  gr_1,gr_2,gr_3,gr_4, i;
 28:
 29:          ap_id = appl_init();
 30:          handle = graf_handle(&gr_1,&gr_2,&gr_3,
                                    &gr_4);
 31:          for (i=0; i<10; work_in[i++]=1)
 32:              ;
 33:          work_in[10]=2;
 34:          v_opnvwk(work_in, &handle, work_out);
 35:  }
 36:
 37:  void gem_exit()     /* Gegenstück zu gem_init()
 38:                         am Ende der Applikation */
 39:  {
 40:          v_clsvwk(handle);
 41:          appl_exit();
 42:          exit(0);
 43:  }
 44:
 45:  void
 46:  clear_window(w_hndl) /* loescht Fensterinhalt  */
 47:    WORD w_hndl;
 48:  {
 49:    WORD  clip[4];
 50:
 51:    wind_get(w_hndl,WF_WORKXYWH,
 52:            &clip[0],&clip[1],&clip[2],&clip[3]);
 53:    clip[2] += clip[0]; clip[2]--;
 54:    clip[3] += clip[1]; clip[3]--;
 55:    v_hide_c(handle);
 56:    vsf_color(handle,0);
 57:    vs_clip(handle,1,clip);
 58:    v_bar(handle,clip);
 59:    v_show_c(handle);
 60:  }
 61:
 62:  /* erweiterte open_window()-Funktion ... */
 63:
 64:  WORD open_window(w_name,was,algn,snp,x1,y1,w1,h1,
 65:                                         mx,my,mw,mh)
 66:    char     *w_name;      /* Ptr auf Namensstring */
 67:    WORD     was,          /* Liste der Elemente    */
 68:             algn,snp,     /* align-Wert, snap j/n */
 69:             x1,y1,w1,h1,  /* Startkoordinaten */
 70:             mx,my,mw,mh;  /* Maximalkoordinaten */
 71:                           /* falls mw==0 gleich */
 72:                           /* Desktop */
 73:  {
 74:    WORD     w_handle;
 75:
 76:    if(mw==0)
 77:      /* ermittelt die Größe des Desktop: */
 78:      wind_get(0,4,&mx,&my,&mw,&mh);
 79:
 80:      /* und meldet ein Fenster an */
 81:    w_handle=wind_create(was,mx,my,mw,mh);
 82:
 83:      /* trägt wichtige Daten in Struktur ein : */
 84:    windows[w_handle].max.g_x  = mx;
 85:    windows[w_handle].max.g_y  = my;
 86:    windows[w_handle].max.g_w  = mw;
 87:    windows[w_handle].max.g_h  = mh;
 88:    windows[w_handle].elements = was;
 89:    windows[w_handle].align    = algn;
 90:    windows[w_handle].snap     = snp;
 91:    strcpy(windows[w_handle].name,w_name);
 92:    windows[w_handle].full =
 93:         (x1==mx && y1==my && w1==mw && h1==mh);
 94:
 95:      /* setzen des Fensternamens : */
 96:    wind_set(w_handle,2,ADDR(windows[w_handle].name)
 97:           ,0,0);
 98:      /* zeichnet öffnende Box:              */
 99:    graf_growbox(0,0,0,0,x1,y1,w1,h1);
100:
101:    wind_open(w_handle,x1,y1,w1,h1);
102:    clear_window(w_handle);
103:
104:      /* schliesslich geben wir dem rufenden
105:         Programm die Identifikationsnummer
106:         des Fensters zurück:             */
107:    return(w_handle);
108:  }
109:
110:
111:
112:
113:  WORD               /* vergleicht zwei GRECT- */
114:    rc_equal(p1,p2) /* Strukturen auf Gleichheit */
115:    GRECT    *p1,*p2;
116:  {
117:    if((p1->g_x != p2->g_x) ||
118:       (p1->g_y != p2->g_y) ||
119:       (p1->g_w != p2->g_w) ||
120:       (p1->g_h != p2->g_h)) return(FALSE);
121:    return(TRUE);
122:  }
123:
124:  void
125:  handle_full(w_hndl)
126:    WORD  w_hndl;
127:  {
128:    GRECT prev;
129:    GRECT curr;
```

→

```
130:    GRECT full;
131:
132:    /* ermittelt die momentane, vorherige und
133:       maximale Größe des Fensters : */
134:    wind_get(w_hndl,WF_CURRXYWH,&curr.g_x,&curr.g_y
135:                              ,&curr.g_w,&curr.g_h);
136:    wind_get(w_hndl,WF_PREVXYWH,&prev.g_x,&prev.g_y
137:                              ,&prev.g_w,&prev.g_h);
138:    wind_get(w_hndl,WF_FULLXYWH,&full.g_x,&full.g_y
139:                              ,&full.g_w,&full.g_h);
140:    if (rc_equal(&curr, &full))
141:    { /* Window ist auf voller Größe, jetzt auf
142:         alte Größe setzen :  */
143:      graf_shrinkbox(prev.g_x,prev.g_y,
144:              prev.g_w,prev.g_h,full.g_x,full.g_y,
145:                              full.g_w,full.g_h);
146:      wind_set(w_hndl,WF_CURRXYWH,prev.g_x,prev.g_y,
147:                              prev.g_w,prev.g_h);
148:      windows[w_hndl].full = FALSE;
149:    }
150:    else
151:    { /* Window ist nicht auf voller Größe,
152:         deshalb auf volle Größe setzen :       */
153:      graf_growbox(curr.g_x,curr.g_y,
154:              curr.g_w,curr.g_h,full.g_x,full.g_y,
155:                              full.g_w,full.g_h);
156:      wind_set(w_hndl,WF_CURRXYWH,full.g_x,full.g_y,
157:                              full.g_w,full.g_h);
158:      windows[w_hndl].full = TRUE;
159:    }
160:  }
161:
162:  /* Routine, die dafür sorgt, daß ein Fenster */
163:  /* vollständig innerhalb eines spezifizierten */
164:  /* Rechtecks liegt (z.B. Desktop)   :        */
165:  void
166:  snap(w1,w2)
167:    GRECT *w1,*w2;
168:  {
169:    if(w2->g_x < w1->g_x) w2->g_x = w1->g_x;
170:
171:    if(w2->g_y < w1->g_y) w2->g_y = w1->g_y;
172:
173:    if((w2->g_x + w2->g_w) > (w1->g_x + w1->g_w))
174:      w2->g_x = (w1->g_x + w1->g_w) - w2->g_w;
175:
176:    if((w2->g_y + w2->g_h) > (w1->g_y + w1->g_h))
177:      w2->g_y = (w1->g_y + w1->g_h) - w2->g_h;
178:
179:    if(w2->g_x < w1->g_x)
180:    {  w2->g_x = w1->g_x; w2->g_w = w1->g_w; }
181:
182:    if(w2->g_y < w1->g_y)
183:    {  w2->g_y = w1->g_y; w2->g_h = w1->g_h; }
184:
185:
186:
187:  }
188:
189:
190:  /* Routine, die die Koordinate k auf ein        */
191:  /* Vielfaches von v "snappt" :                  */
192:  WORD
193:  align(k,n)
194:    WORD  k,n;
195:  {
196:    k += (n>>1) - 1;
197:    k  = n * (k / n);
198:    return(k);
199:  }
200:
201:  void              /* allg. Fensterverwaltung    */
202:  handle_window(buffer)
203:    WORD  buffer[8];
204:  {
205:    WORD algn;
206:
207:    switch(buffer[0])
208:    {
209:      case MN_SELECTED: break;
210:      case WM_REDRAW   : clear_window(buffer[3]);
211:                         break;
212:      case WM_TOPPED   : wind_set(buffer[3],WF_TOP,
213:                         buffer[3],0,0,0);
214:                         break;
215:      case WM_CLOSED   : wind_close(buffer[3]);
216:                         break;
217:      case WM_FULLED   : handle_full(buffer[3]);
218:                         break;
219:      case WM_ARROWED :
220:          form_alert(1,"[1][WM_ARROWED ...][ok]");
221:          break;
222:      case WM_HSLID    :
223:                       wind_set(buffer[3],WF_HSLIDE,
224:                               buffer[4],0,0,0);
225:                       break;
226:      case WM_VSLID    :
227:                       wind_set(buffer[3],WF_VSLIDE,
228:                               buffer[4],0,0,0);
229:                       break;
230:      case WM_SIZED    :
231:      case WM_MOVED    :
232:      /* Wenn Fenster auf maximaler Größe und
233:         Breite bzw. Höhe vergrößert -> Abbruch */
234:                       if(windows[buffer[3]].full
235:          && ( buffer[6]>windows[buffer[3]].max.g_w
236:             || buffer[7]>windows[buffer[3]].max.g_h))
237:                         break;
238:      /* Wenn kleiner Minimalgröße vergrößern: */
239:                       if(buffer[6]<133)
                              buffer[6] = 133;
240:                       if(buffer[7]<133)
                              buffer[7] = 133;
241:      /* Ausrichten der X-Koordinate, falls algn
242:         nicht gleich 0 */
243:                       algn=windows[buffer[3]].align;
244:                       if(algn)
245:                         buffer[4] =
246:                         align(buffer[4],algn)-1;
247:
248:      /* Falls Fenster außerhalb Maximalkoordinaten
249:         wieder zurückschieben : */
250:                       if(windows[buffer[3]].snap)
251:                       snap(&windows[buffer[3]].max,
252:                             &buffer[4]);
253:
254:      /* Mit den neu berechneten Koordinaten das
255:         Fenster neu positionieren : */
256:                       wind_set(buffer[3],WF_CURRXYWH,
257:                               buffer[4],buffer[5],
258:                               buffer[6],buffer[7]);
259:                       break;
260:      case WM_NEWTOP   : clear_window(buffer[3]);
261:                         break;
262:      default          : break;
263:    }
264:  }
```

```
 1:       /* Demonstrationsprogramm zum Snappen  */
 2:      /* von Windows                          */
 3:     /* Andreas Loetscher 1988               */
 4:    /* Compiler : Lattice-C 3.04            */
 5:   /****************************************/
 6:
 7:  #include <a:\headers\portab.h>
 8:  #include <c:\listing1.h>
 9:
10:  void
11:  main()
12:  {
13:    WORD  buffer[8], w_handle;
14:
15:    gem_init();
16:
17:    w_handle = open_window("Demofenster",
18:                           NAME+CLOSER+FULLER+
19:                           MOVER+SIZER,
20:                           20,TRUE,
21:                           40,30,300,300,
22:                           0,19,600,350);
23:
24:    do
25:    {
26:      evnt_mesag(buffer);
27:      handle_window(buffer);
28:    }while(buffer[0] != WM_CLOSED);
29:
30:    wind_delete(w_handle);
31:    gem_exit();
32:  }
```

Lovely Helper

Ein Desk-Accessory 4

Einige Probleme rund um den Kalender

In der vierten Folge des Lovely Helpers wird es heute um ein Problem gehen, daß die Menschheit schon seit gut 2000 Jahren beschäftigt. Ich spreche von den mehr oder weniger fruchtbaren Bemühungen einiger Generationen, ihre Zeitrechnung an die physikalische Länge eines Jahres anzupassen.

Aus dem Lexikon ...

... können wir dazu entnehmen,

"[...] [daß] maßgebend für unseren Kalender das von Caesar 46 v.C. eingeführte feste Sonnenjahr [ist]. Danach hat ein Jahr 365 Tage, jedes vierte als Schaltjahr 366 Tage (Julianischer Kalender); da aber die mittlere Jahreslänge von 365.25 Tagen gegen das Sonnenjahr um 11 Minuten 12 Sekunden zu groß ist, führte Papst Gregor XIII 1582 ein genaueres Schaltverfahren ein, bei dem der Schalttag beim vollen Jahrhundert mit Ausnahme der durch 4 teilbaren ausfällt (Gregorianischer Kalender). [...]"

Was die Menschheit eine gewisse Zeit lang in Atem gehalten hat, können wir in Pascal in wenigen Zeilen notieren. Zunächst ist da das Schaltjahr. Wenn wir obiges Zitat ein bißchen sortieren, kommt dabei folgendes heraus:

- Jedes vierte Jahr ist ein Schaltjahr.
- Alle 100 Jahre fällt dieses Schaltjahr aus.
- Alle 400 Jahre findet es trotzdem statt.

In Pascal ergibt sich - kurz und bündig - die Routine schaltjahr (Listing 9, Zeilen 46-52). Sie ermittelt für jedes beliebige Jahr, ob ein Schaltjahr vorliegt oder nicht. Einige einfache, abgeleitete Routinen sind:

anzahl_tage (Zeilen 54-69) - diese Integerfunktion berechnet für jeden Monat jeden Jahres die Anzahl seiner Tage.

datum_korrekt (Zeilen 81-93) - eine boolsche Funktion, die zu gegebenem Datum ermittelt, ob es sich dabei um ein zulässiges Datum handelt oder nicht.

Anmerkung: Daten vor 1582 sind auf gar keinen Fall zulässig, da wir uns ausschließlich auf den Gregorianischen Kalender beziehen, der erst ab 1582 gültig ist. Wenn Sie das stört, können Sie die Sperre in *datum_korrekt* ja ausbauen. Allerdings: Sinnvoll ist das nicht!

Eine Ordinalfunktion für Kalenderdaten

Sagt Ihnen folgende Pascal-Funktion etwas?

```
FUNCTION to_lower(ch : char) : char;
BEGIN
  to_lower:=chr(ord(ch)-ord('A')+ord('a'));
END;
```

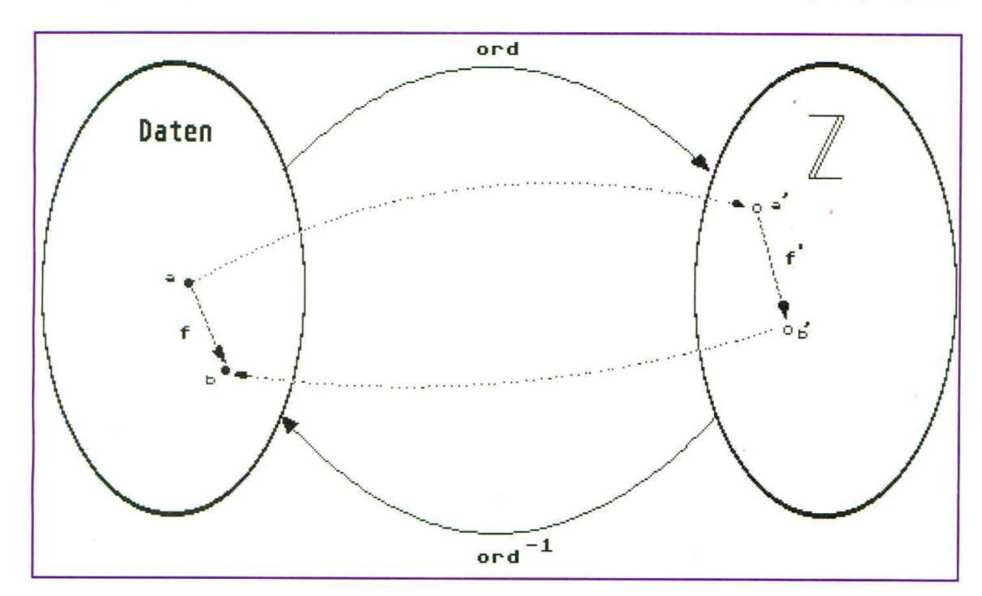
Es ist eine recht elegante Umwandlung von Großbuchstaben in Kleinbuchstaben, die sich eine der hervorstechendsten Eigenschaften des Datentyps Char zu Nutzen macht: Char ist ein ordinaler Datentyp. Dies bedeutet, daß für alle Elemente des Datentyps Char eine Standardfunktion - namentlich ord - existiert, die jedem Element in eindeutiger Weise einen Integerwert zuweist. Umgekehrt existiert für jeden so berechenbaren Integerwert die Standardfunktion chr, welche wieder den zugehörigen Charakter ermittelt. Damit ist es möglich, Elemente aus Char zunächst nach Integer (ord(ch)) abzubilden, dort eine Berechnung auszuführen (+,- ,...) und anschließend wieder nach Char zu gelangen *[chr(...)]*.

Im obigen Beispiel haben wir so zunächst die Variable ch und die beiden Konstanten 'A' und 'a' nach Integer abgebildet, den Offset für Großbuchstaben ord('A') von ord(ch) abgezogen und anschließend den Offset für Kleinbuchstaben ord('a') wieder hinzuaddiert. Mit dem Ergebnis, daß wir den ch zugehörigen Kleinbuchstaben ermittelt haben.

Jetzt werden Sie sich vielleicht fragen, was das denn alles mit dem Kalender zu tun hat? Die Antwort ist ganz einfach: Gelingt es uns, für Daten ebenfalls eine Ordinalfunktion und ihr Inverses zu ermitteln, dann sind wir fein raus, weil wir dann auch mit Daten (!) rechnen können.

Nun - ich darf es wohl vorwegnehmen -, natürlich gelingt uns die Konstruktion

einer derartigen Ordinalfunktion und ihrer Inversen. Bevor ich aber die Programmierung derselben bespreche, zunächst zurück zu dem Nutzen, den wir dadurch erlangen. Betrachten Sie bitte die Abbildung 13. Dargestellt ist eine Ordinalfunktion von den Daten in die ganzen Zahlen und ihr Inverses. Will man in den Daten eine Funktion f ausführen, etwa die Differenz zweier Daten, so ist es unnötig, dazu eine eigene Operation zu definieren. Man



erlangt f auch einfach, indem man zunächst mit ord in die ganzen Zahlen geht, dort die Funktion f' ausführt und dann mit ord-1 wieder zurück in die Daten gelangt. Notiert als mathematische Funktion sieht dies folgendermaßen aus:

f(a) = ord-1(f'(ord(a))) = b

Doch lassen wir die Mathematik beiseite, sonst fallen mir noch so Begriffe wie Homomorphismus ein. Das möchte ich Ihnen doch ersparen. Natürlich können wir auch ohne derart hochmathematisches Zeug eine Ordinalfunktion für unsere Kalenderdaten sowie ihr Inverses angeben. Geschehen ist dies mit den beiden Prozeduren *ord_datum* (Ordinalfunktion, Zeilen 95-109) und *datum_ord* (Inverse, Zeilen 111-138).

Die einfachere von beiden ist die Ordinalfunktion selber. Hier berechnen wir nur die Anzahl der Tage vom Beginn der Zeitrechnung bis zum eingegebenen Datum, indem wir die verstrichenen Jahre/Schaltjahre, Monate und Tage in der Einheit "Tag" einfach aufaddieren.

Auch umgekehrt ist die Berechnung nicht wesentlich schwerer. Wir gehen so vor, daß zunächst das Jahr, dann der Monat und dann der Tag aus dem Ordinalwert (übrigens ein *Long_integer*) zurückberechnet wird. Hier ist allerdings zu beachten, daß wir für die Variable jahr erst nur einen Näherungswert ermitteln, der dann - nach Ermittlung der Schaltjahre - eventuell nach unten korrigiert werden muß (Zeilen 123-130). Von der verbleibenden Anzahl Tage werden nun, beginnend beim Monat Januar, solange Monate abgezogen, bis nicht mehr genug Tage für einen vollen Monat übrig sind (Zeilen 131-136). Als Wert für den gesuchten Monat erhält man dann die Anzahl der Schleifendurchläufe; der gesuchte Tag ist der Rest des Ordinalwertes. Nun zwei Beispiele für Berechnungen mit diesen beiden Funktionen:

1. Wieviele Tage liegen zwischen Beginn und Ende unseres Jahrhunderts?

```
ord_datum(1,1,1900,ord1);
ord_datum(31,12,1999,ord2);
writeln(ord2-ord1);
```

Antwort: 36523

2. *1989 ist Ostern am 26.3., Pfingsten ist sieben Wochen nach Ostern. An welchem Tag ist 1989 Pfingsten?*

```
tag:=26;
monat:=3;
jahr:=1989;
ord_datum(tag,monat,jahr,ord1);
datum_ord(ord1+49,tag,monat,jahr);
writeln(tag:2,'-',monat:2,'-',jahr:4);
```

Antwort: 14- 5-1989

Genauso einfach erhalten wir den Wochentag eines beliebigen Datums (*wochentag_berechnen*, Zeilen 140-149). Kurz gesagt wird nur der Ordinalwert des Datums modulo 7 gerechnet. Das Resultat - irgendetwas zwischen 0 und 6 - kann gemäß Prozedur tagesname (Zeilen 171-184) interpretiert werden.

Beispiel:

Was für ein Tag war der Geburtstag Goethes (28.8.1749)?

```
writeln(wochentag_berechnen(28,8,1749));
```

Antwort: 4 (==> Donnerstag)

Feiertage

Als nächstes wollen wir uns mit der Thematik "Feiertage" beschäftigen, wozu Sie Ihre Aufmerksamkeit bitte zunächst der Tabelle 1 zuwenden. Aufgeführt sind sämtliche Feiertage, die entweder bundesweit anerkannt sind oder nur lokal für einige Bundesländer gelten.

In Pascal definieren wir einen Aufzählungstyp zur Darstellung all dieser Feiertage (*ft_type*, Zeilen 19-23). Ebenso verfahren wir mit den unterschiedlichen Bundesländern, da sie ebenfalls zur genauen Feiertagsbestimmung herangezogen werden müssen (bundesland, Zeilen 15-17).

Wenn wir jetzt einen tieferen Blick in die Tabelle 1 werfen, so lassen sich insgesamt drei unterschiedliche Gruppen von Feiertagen erkennen:

1. die festen Feiertage - sie finden immer am gleichen Tag innerhalb eines Jahres statt.

2. die Feiertage, die sich aus ihrer Abhängigkeit vom Osterdatum ergeben.

3. der "Buß- und Bettag" mit einer separaten Berechnungsformel.

Die einzige komplexere Operation ist dabei die Berechnung des Osterdatums. Wie Sie der Tabelle 1 entnehmen können, fällt dieser Feiertag gerade immer auf den ersten Sonntag nach dem ersten Frühlingsvollmond. Da größere Mondbeobachtung zur Ermittlung dieser Formel aus Zeitgründen bei mir leider ausscheiden, war ich genötigt eine kleine Anleihe bei einem älteren "BASIC-Source" zu tätigen.

Man sieht es der Routine osterdatum (Zeilen 204-254) geradezu an: keine Spur von strukturierter Programmierung! Benutzen wir diese Formel also besser als Black-Box, damit ich bei der Besprechung nicht allzusehr ins Schwitzen gerate. Mithin ergibt sich Prozedur *datum_*

ft_berechnen (Zeilen 256-351). In dieser (nun) recht einfachen Prozedur werden für einen beliebigen Feiertag innerhalb eines Jahres das genaue Datum sowie seine Bezeichnung berechnet. In der oberen Hälfte des zugehörigen CASE-Statements finden sich dabei die einfachen Feiertage mit den konstanten Daten.

Interessant ist vielleicht noch die Berechnung des "Buß- und Bettags" (Zeilen 310-320). Ausgehend vom "Heiligen Abend" wird hier der "4. Advent" berechnet, indem, am 24.12 beginnend, langsam die Tageszahl runtergezählt wird, bis ein Sonntag erreicht ist. Dieser ist dann per Definition der "4. Advent". Von da an sind nur noch 32 Tage abzuziehen, was - mit unseren bisherigen Operationen - sehr einfach ist. Alle übrigen Feiertage ergeben sich durch Subtraktion bzw. Addition einer bestimmten Tageszahl zum Osterdatum.

Die nächste größere Operation ist die Prozedur *feld_ft_berechnen* Zeilen 353-447). Sie erhält als Parameter eine Jahreszahl und ein Bundesland und hat als Ergebnis ein nach dem Datum sortiertes Feld von Feiertagen (*ft_feld*, Zeilen 31-34) zu berechnen. Die Feldeinträge besitzen dabei jeweils eine Feiertagsbeschreibung in Form eines Strings sowie das genaue Datum des Feiertags. In einer FOR-Schleife (Zeile 437-443) über alle Feiertage wird zunächst ermittelt, ob der indizierte Feiertag auch in dem betrachteten Bundesland gültig ist. Ist dies der Fall, werden mit Hilfe von *datum_ft_ berechnen* die zugehörigen Daten für das *ft_feld* ermittelt. Am Ende der FOR-Schleife stehen nun sämtliche Feiertage im *ft_feld*, die im betrachteten Bundesland <land> Gültigkeit haben. Was bleibt, ist die Sortierung der Daten.

Bezeichnung:	Termin:	
Neujahrstag	1. 1	
Heilige Drei Könige	6. 1 [1)]	
Maifeiertag	1. 5	
Tag der deutschen Einheit	17. 6	
Mariä Himmelfahrt	15. 8 [2)]	
Allerheiligen	1.11 [3)]	
1. Weihnachtstag	25.12	
2. Weihnachtstag	26.12	
Buß- und Bettag	32 Tage vor dem 4. Advent	
Karfreitag	2 Tage vor Ostern	
Ostersonntag	Sonntag auf den 1.	Frühlingsvollmond
Ostermontag	1 Tag nach Ostern	
Christi Himmelfahrt	39 Tage nach Ostern	
Pfingstsonntag	49 Tage nach Ostern	
Pfingstmontag	50 Tage nach Ostern	
Fronleichnam	60 Tage nach Ostern [4)]	

Tabelle 1: Bundesweite & lokale Feiertage

Kalender Mai 1989

1	8	**15**	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	**25**	
5	12	19	26	
6	13	20	27	
7	**14**	**21**	**28**	

Gesetzliche Feiertage in Nordrhein-Westfalen:

1.5 Maifeiertag
4.5 Christi Himmelfahrt
14.5 Pfingstsonntag
15.5 Pfingstmontag
25.5 Fronleichnam

Abb. 14: Ein Beispiel für den Monatsdruck

Daß ich zur Sortierung dieses recht kleinen Arrays Quicksort verwendet habe, könnte böse Zungen unter Ihnen veranlassen, mit dem Sprichwort von den Spatzen und den Kanonen aufzuwarten. Zu meiner Verteidigung sei jedoch gesagt, daß Quicksort nicht nur der effektivste bekannte Sortieralgorithmus ist, sondern zugleich auch einer der einfachsten. Außerdem werden wir Quicksort im Laufe eines der nächsten Helperteile noch einmal bemühen, dann allerdings mit etwas mehr Berechtigung als heute.

Quicksort

Für diejenigen unter Ihnen, die Quicksort noch nicht kennen, hier noch eine kurze Erläuterung seiner Funktionsweise, die ich - bis auf Kommentierungen und Zeilenangaben - aus meinem Buch über Datenstrukturen zitiere. Man soll ja schließlich das Rad nicht immer wieder neu erfinden:

Der Quicksortalgorithmus, entwickelt von C.A.R. Hoare, ist bis zum heutigen Tage DER Sortieralgorithmus. Obwohl er sehr einfach und somit schnell zu beschreiben ist, ist er, in der Praxis, die beste bekannte Sortiermethode.

Erster Schritt des Quicksortalgorithmus' ist die Auswahl eines beliebigen Arrayelementes, das wir als Pivotelement, oder kurz pivot, bezeichnen wollen (Zeile 372-373). Alle Arrayelemente kleiner diesem Element werden dann auf die linke Seite des Arrays befördert, alle größeren Elemente auf die rechte Seite. Aus einem großen Array entstehen dadurch zwei Teilarrays, a[1]..a[j] und a[i]..a[max], für die gilt:

a[x]<=a[y] , für alle x aus 1..j
und alle y aus i..max, wobei j=i-1 ist.

Wir haben also praktisch durch diesen Austauschvorgang die Sortierung eines großen Arrays auf die Sortierung zweier kleiner Arrays reduziert. Durch Anwendung von Quicksort auf die so entstehenden Teilarrays erreicht man, indem man die Arraylänge bis auf eins herunterspielt, letztendlich ein sortiertes Gesamtarray.

Welches Arrayelement als Pivotelemente gewählt wird, ist zwar bei einem zufällig erzeugten Array vollständig belanglos, bei "normalen" Arrays, die üblicherweise in irgendeiner Form bereits eine Teilsortierung aufweisen, hat es sich als sinnvoll erwiesen, genau das mittlere Element des Teilarrays (l..r) auszuwählen.

Die REPEAT-Schleife realisiert nun die Zerlegung des Teilarrays. Zu diesem Zweck werden zwei Zeiger, i und j, von links und rechts über das Array geschoben (die beiden WHILE-Schleifen, Zeilen 378-388), und bei Bedarf werden zwei Elemente ausgetauscht.

Nach Abbruch dieses Vorgangs sind nun nur noch zwei rekursive Aufrufe von quicksort, mit den neuen Teilarraygrenzen l..j und i..r, zu tätigen, natürlich nur, soweit es sich dabei noch um echte Teilarrays handelt, also l<j, bzw. i<r gilt (Zeile 399-402).

Anmerkung:
Bei der heutigen Realisierung von Quick sort benutzen wir anstelle der üblichen <= Relation, die <= Relation, angewendet auf die durch unsere Ordinalfunktion ermittelten Werte. Vor Auswertung der Relation ist also jedesmal ein Aufruf von *ord_datum* mit dem Datum des betroffenen Arrayelementes zu tätigen (Zeilen 377, 381, 383 und 387).

Die Berechnung einer Monatsmatrix

Die endgültig letzte Operation, die wir uns heute vornehmen werden, ist die Berechnung einer 6x7-Matrix (*monats_matrix*, Zeile 41), die sämtliche Tage eines Monats vorformatiert aufnehmen kann. Wie bei Kalendern üblich, sollen die Montage dazu in die erste Zeile geschrieben werden, die Dienstage in die zweite und so fort. Für alle Sonn- und Feiertage soll ein Flag (fett) aufgenommen werden, das den Sonderstatus dieser Tage charakterisiert. Gehen wir es an.

Zunächst erfolgt in *berechne_matrix* eine Initialisierung dieser *monats_matrix* (Zeilen 474-479). Danach wird in der ersten Spalte mit dem Eintrag aller Tage des Monats begonnen. Zu jedem Tag wird dazu zunächst der Wochentag berechnet und der entsprechende Matrixeintrag vorgenommen. Liegt ein Sonntag vor, kann das Feiertagsflag auch gleich gesetzt werden. Zusätzlich rückt der Spaltenzähler um eine Position vor, da mit dem nächsten Wochentag die Matrixkomponenten in die nächste Spalte geschrieben werden müssen. Bei einem anderen Wochentag wird mit der Funktion *is_feiertag* nachgesehen, ob für diesen Tag ein Eintrag im *ft_feld* existiert. Wenn ja, ist hier ebenfalls das Feiertagsflag zu setzen. Resultat von *berechne_matrix* ist eine Variable, mit der es uns in der nächsten Folge auf sehr einfache Weise möglich sein wird, einen Monat auszudrucken.

Vorausschau

Als kleinen Vorgeschmack darauf können Sie schon einmal die Abbildung 14 betrachten. Genauso wird der Ausdruck eines Monats nämlich aussehen. Darüber hinaus wird auch noch der Ausdruck eines kompletten Jahres implementiert werden. Auch unser Utility zum Setzen und Anzeigen der Systemzeit werden wir in der nächsten Folge fertigbekommen. In der Hoffnung, daß es Ihnen bei meinem Helper immer noch nicht langweilig geworden ist, verbleibe ich bis dahin

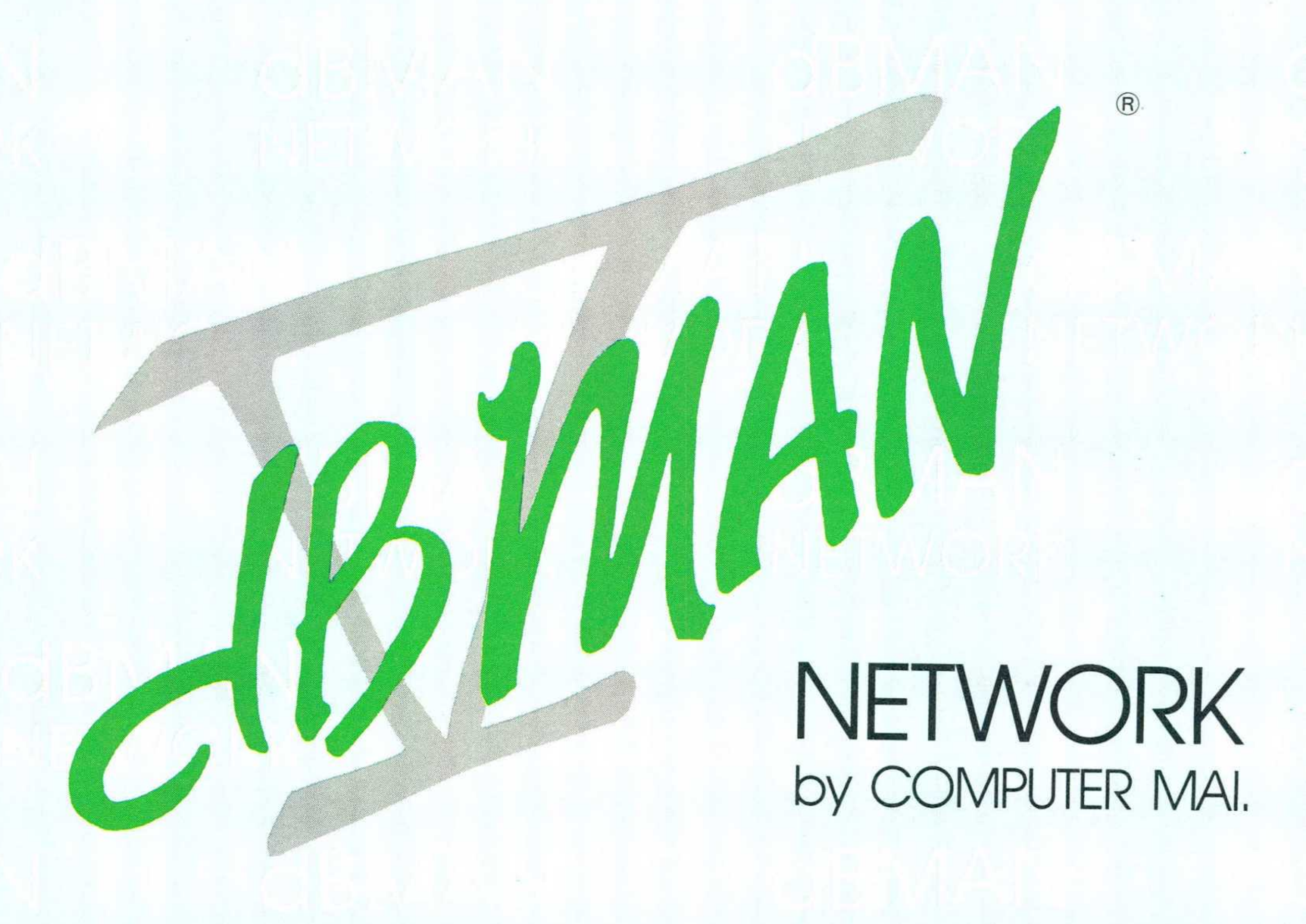
D.Brockhaus

```
 1:  {**********************************************}
 2:  {* Listing 09   : Nicht zu GEM gehörige
                       Zeitgrundfunktionen z.B.:  *}
 3:  {*   1. Die Ordinalfunktionen zum Datum       *}
 4:  {*   2. Berechnung von festen und beweglichen *}
 5:  {*      Feiertagen                            *}
 6:  {*   3. Berechnung einer Monatsmatrix, in der *}
 7:  {*      Sonn- und Feiertage gekennzeichnet sind.*}
 8:  {*                                             *}
 9:  {* Datei        : ZEIT1.PAS                    *}
10:  {* last update  : 30.4.1988                    *}
11:  {**********************************************}
12:
13:  CONST max_feiertage = 16;
14:
15:  TYPE bundesland =(schleswig_holstein,hamburg,
                       bremen,niedersachsen,
16:                    nordrhein_westfalen,hessen,
                       rheinland_pfalz,
17:                    saarland,baden_wuertemberg,
                       bayern);
18:
19:       ft_type = (neujahr,karfreitag,ostersonntag,
                     ostermontag,
20:                  maifeiertag,christi_himmelfahrt,
                     pfingstsonntag,
21:                  pfingstmontag,deutsche_einheit,
                     buss_und_bettag,
22:                  weihnach_1,weihnach_2,
                     hl_drei_koenige,fronleichnam,
23:                  mariae_himmelfahrt,allerheiligen);
24:
25:       ft_besch = RECORD
26:                    tag    ,
27:                    monat  : integer;
28:                    bez    : string;
29:                  END;
30:
31:       ft_feld = RECORD
32:                   anz : 1..max_feiertage;
33:                   a   : ARRAY [1..max_feiertage]
                            OFft_besch;
34:                 END;
35:
36:       tages_type = RECORD
37:                      tag  : integer;
38:                      fett : boolean;
39:                    END;
40:
41:       monats_matrix = ARRAY [1..6,1..7]
                          OFtages_type;
42:
43:  VAR default_monat    ,
44:      default_jahr     : integer;
45:
46:  FUNCTION schaltjahr (jahr : integer) : boolean;
47:
48:    BEGIN
49:      schaltjahr:=((jahr MOD 400=0) OR
50:                  NOT (jahr MOD 100=0)) AND
51:                  (jahr MOD 4=0)
52:    END;
53:
54:  FUNCTION anzahl_tage(monat ,
55:                       jahr   : integer) : integer;
56:
57:    VAR tage : integer;
58:
59:    BEGIN
60:      CASE monat OF                                  →
```

```
61:         4,6,9,11 : tage:=30;
62:         2        : IF schaltjahr(jahr) THEN
63:                      tage:=29
64:                    ELSE
65:                      tage:=28;
66:         OTHERWISE: tage:=31;
67:       END;
68:       anzahl_tage:=tage;
69:     END;
70:
71:   FUNCTION uhrzeit_korrekt(stunde  ,
72:                            minute  ,
73:                            sekunde : integer):
                               boolean;
74:
75:     BEGIN
76:       uhrzeit_korrekt:=(stunde IN [0..23]) AND
77:                        (minute IN [0..59]) AND
78:                        (sekunde IN [0..59]);
79:     END;
80:
81:   FUNCTION datum_korrekt(tag   ,
82:                          monat ,
83:                          jahr  : integer) :
                             boolean;
84:
85:     VAR tage : integer;
86:
87:
88:     BEGIN
89:       tage:=anzahl_tage(monat,jahr);
90:       datum_korrekt:=(tag IN [1..tage]) AND
91:                      (monat IN [1..12]) AND
92:                      (jahr >= 1582);
93:     END;
94:
95:   PROCEDURE ord_datum(    tag   ,
96:                           monat ,
97:                           jahr  : integer;
98:                       VAR ordnr : long_integer);
99:
100:    VAR i     : integer;
101:        lyear : long_integer;
102:
103:    BEGIN
104:      lyear:=pred(jahr);
105:      ordnr:=lyear * 365 + lyear DIV 4 - lyearDIV
100 +
106:             lyear DIV 400 + tag;
107:      FOR i:=1 TO monat-1 DO
108:        ordnr:=ordnr+anzahl_tage(i,jahr);
109:    END;
110:
111:  PROCEDURE datum_ord(    ordnr : long_integer;
112:                      VAR tag   ,
113:                          monat ,
114:                          jahr  : integer);
115:
116:    VAR lyear : long_integer;
117:
118:    BEGIN
119:      jahr:=ordnr DIV 365 + 1;
120:      lyear:=pred(jahr);
121:      ordnr:=ordnr - lyear * 365 - lyear DIV 4 +
122:             lyear DIV 100 - lyear DIV 400;
123:      WHILE ordnr<=0 DO
124:        BEGIN
125:          jahr:=pred(jahr);
126:          IF schaltjahr(jahr) THEN
127:            ordnr:=ordnr+366
128:          ELSE
129:            ordnr:=ordnr+365;
130:        END;
131:      monat:=1;
132:      WHILE ordnr>anzahl_tage(monat,jahr) DO
133:        BEGIN
134:          ordnr:=ordnr-anzahl_tage(monat,jahr);
135:          monat:=succ(monat);
136:        END;
137:      tag:=ordnr;
138:    END;
139:
140:  FUNCTION wochentag_berechnen(tag   ,
141:                               monat ,
142:                               jahr  : integer ):
                                  integer;
143:
144:    VAR ordnr : long_integer;
145:
146:    BEGIN
147:      ord_datum(tag,monat,jahr,ordnr);
148:      wochentag_berechnen:=int(ordnr MOD 7);
149:    END;
150:
151:  PROCEDURE monatsname(    nr   : integer;
152:                       VAR text : string);
153:
154:    BEGIN
155:      CASE nr OF
156:        1 : text:='Januar';
157:        2 : text:='Februar';
158:        3 : text:='März';
159:        4 : text:='April';
160:        5 : text:='Mai';
161:        6 : text:='Juni';
162:        7 : text:='Juli';
163:        8 : text:='August';
164:        9 : text:='September';
165:        10: text:='Oktober';
166:        11: text:='November';
167:        12: text:='Dezember';
168:      END;
169:    END;
170:
171:  PROCEDURE tagesname(    nr   : integer;
172:                      VAR text : string);
173:
174:    BEGIN
175:      CASE nr OF
176:        0 : text:='So';
177:        1 : text:='Mo';
178:        2 : text:='Di';
179:        3 : text:='Mi';
180:        4 : text:='Do';
181:        5 : text:='Fr';
182:        6 : text:='Sa';
183:      END;
184:    END;
185:
186:  PROCEDURE landesname(    land : bundesland;
187:                       VAR text : string);
188:
189:    BEGIN
190:      CASE land OF
191:        schleswig_holstein  :
                              text:='Schleswig-Holstein';
192:        hamburg             :text:='Hamburg';
193:        bremen              :text:='Bremen';
194:        niedersachsen       :text:='Niedersachsen';
195:        nordrhein_westfalen :
                              text:='Nordrhein-Westfalen';
196:        hessen              :text:='Hessen';
197:        rheinland_pfalz     :
                              text:='Rheinland-Pfalz';
198:        saarland            :text:='Saarland';
199:        baden_wuertemberg   :
                              text:='Baden-Würtemberg';
200:        bayern              :text:='Bayern';
201:      END;
202:    END;
203:
204:  PROCEDURE osterdatum(    jahr  : integer;
205:                       VAR tag   ,
206:                           monat : integer);
207:
208:    VAR p ,
209:        n ,
210:        q ,
211:        r ,
212:        x ,
213:        y ,
214:        a ,
215:        b ,
216:        c ,
217:        d ,
218:        e : integer;
219:
220:    BEGIN
221:      p:=jahr DIV 100;
222:      n:=jahr MOD 100;
223:      q:=p DIV 3;
224:      r:=p DIV 4;
```

→

dBMAN NETWORK



dBMAN NETWORK KOSTET NUR 399,– DM dBMAN NETWORK by COMPUTER MAI

Weißenburger Platz 1, 8000 München 80, Tel. 089/4480691, Fax: 089/4483820; Schweiz: ADAG, Züric
Tel. 1/3618323; Österreich: KNEISZ, Wien, Tel. 222-552959; ISYS-COMPUTER, Tel. 0208-65503132.

```
225:      x:=(15+p-q-r) MOD 30;
226:      y:=(p+4-r) MOD 7;
227:      a:=jahr MOD 19;
228:      b:=jahr MOD 4;
229:      c:=jahr MOD 7;
230:      d:=(19*a+x) MOD 30;
231:      e:=(2*b+4*c+6*d+y) MOD 7;
232:      IF 22+d+e<=31 THEN
233:        BEGIN
234:          tag:=22+d+e;
235:          monat:=3;
236:        END
237:      ELSE
238:        IF (d=28) and (e=6) THEN
239:          BEGIN
240:            tag:=18;
241:            monat:=4;
242:          END
243:        ELSE
244:          IF (d=29) and (e=6) THEN
245:            BEGIN
246:              tag:=19;
247:              monat:=4;
248:            END
249:          ELSE
250:            BEGIN
251:              tag:=d+e-9;
252:              monat:=4;
253:            END;
254:    END;
255:
256:  PROCEDURE datum_ft_berechnen(    ft    : ft_type;
257:                                   jahr  : integer;
258:                               VAR tag   ,
259:                                   monat : integer;
260:                               VAR bez   : string);
261:
262:    VAR o_tag   ,
263:        o_monat : integer;
264:        ordnr   : long_integer;
265:
266:    BEGIN
267:      osterdatum(jahr,o_tag,o_monat);
268:      ord_datum(o_tag,o_monat,jahr,ordnr);
269:      CASE ft OF
270:        neujahr              : BEGIN
271:                                 tag:=1;
272:                                 monat:=1;
273:                                 bez:='Neujahrstag';
274:                               END;
275:        hl_drei_koenige      : BEGIN
276:                                 tag:=6;
277:                                 monat:=1;
278:                                 bez:='Heilige drei
                                           Könige';
279:                               END;
280:        maifeiertag          : BEGIN
281:                                 tag:=1;
282:                                 monat:=5;
283:                                 bez:='Maifeiertag';
284:                               END;
285:        deutsche_einheit     : BEGIN
286:                                 tag:=17;
287:                                 monat:=6;
288:                                 bez:='Tag der
                                 deutschen Einheit';
289:                               END;
290:        mariae_himmelfahrt   : BEGIN
291:                                 tag:=15;
292:                                 monat:=8;
293:                                 bez:='Mariä
                                       Himmelfahrt';
294:                               END;
295:        allerheiligen        : BEGIN
296:                                 tag:=1;
297:                                 monat:=11;
298:                                 bez:= 'Aller-
                                          heiligen';
299:                               END;
300:        weihnach_1           : BEGIN
301:                                 tag:=25;
302:                                 monat:=12;
303:                                 bez:='1. Weih-
                                         nachtstag';
304:                               END;
305:        weihnach_2           : BEGIN
306:                                 tag:=26;
307:                                 monat:=12;
308:                                 bez:='2. Weih-
                                         nachtstag';
309:                               END;
310:        buss_und_bettag      : BEGIN
311:                                 tag:=24;
312:                                 monat:=12;
313:                                 WHILE
                             wochentag_berechnen(tag,
314:                                 monat,jahr)<>0 DO
315:                                 tag:=pred(tag);
316:                              ord_datum(tag,monat,
                                          jahr,ordnr);
317:                                ordnr:=ordnr-32;
318:                              datum_ord(ordnr,tag,
                                         monat,jahr);
319:                              bez:='Buß und Bettag';
320:                               END;
321:        karfreitag           : BEGIN
322:                                 datum_ord(ordnr-2,
                                     tag,monat,jahr);
323:                                 bez:='Karfreitag';
324:                               END;
325:        ostersonntag         : BEGIN
326:                                 tag:=o_tag;
327:                                 monat:=o_monat;
328:                                 bez:='Osterso.';
329:                               END;
330:        ostermontag          : BEGIN
331:                                 datum_ord(ordnr+1,
                                     tag,monat,jahr);
332:                                 bez:='Ostermontag';
333:                               END;
334:        christi_himmelfahrt  : BEGIN
335:                                 datum_ord(ordnr+39,
                                     tag,monat,jahr);
336:                                 bez:='Christi Him
                                          melfahrt';
337:                               END;
338:        pfingstsonntag       : BEGIN
339:                                 datum_ord(ordnr+49,
                                     tag,monat,jahr);
340:                                 bez:='Pfingstso.';
341:                               END;
342:        pfingstmontag        : BEGIN
343:                                 datum_ord(ordnr+50,
                                     tag,monat,jahr);
344:                                 bez:='Pfingstmo.';
345:                               END;
346:        fronleichnam         : BEGIN
347:                                 datum_ord(ordnr+60,
                                     tag,monat,jahr);
348:                                 bez:='Fronleich.';
349:                               END;
350:      END;
351:    END;
352:
353:  PROCEDURE feld_ft_berechnen(    land : bundesl.;
354:                                  jahr : integer;
355:                              VAR feld : ft_feld);
356:
357:    VAR i  : ft_type;
358:        nr : integer;
359:
360:    PROCEDURE quicksort(l ,
361:                        r : integer);
362:
363:      VAR i     ,
364:          j     : integer;
365:          help  : ft_besch;
366:          pivot ,
367:          key   : long_integer;
368:
369:      BEGIN
370:        WITH feld DO
371:          BEGIN
372:            ord_datum(a[(l+r) DIV 2].tag,a[(l+r)
                          DIV 2].monat,jahr,
373:                      pivot);
374:            i:=l;
375:            j:=r;
376:            REPEAT
377:              ord_datum(a[i].tag,a[i].monat,jahr,
                            key);
378:              WHILE key<pivot DO                    →
```

```
379:                BEGIN
380:                  i:=succ(i);
381:                  ord_datum(a[i].tag,a[i].monat,
                                          jahr,key);
382:                END;
383:              ord_datum(a[j].tag,a[j].monat,jahr,
                                              key);
384:              WHILE pivot<key DO
385:                BEGIN
386:                  j:=pred(j);
387:                  ord_datum(a[j].tag,a[j].monat,
                                          jahr,key);
388:                END;
389:              IF i<=j THEN
390:                BEGIN
391:                  help:=a[i];
392:                  a[i]:=a[j];
393:                  a[j]:=help;
394:                  i:=succ(i);
395:                  j:=pred(j);
396:                END;
397:            UNTIL i>j;
398:          END;
399:        IF l<j THEN
400:          quicksort(l,j);
401:        IF i<r THEN
402:          quicksort(i,r);
403:      END;
404:
405:    FUNCTION is_feiertag(ft : ft_type) : boolean;
406:
407:      BEGIN
408:        IF ft IN [neujahr..weihnach_2] THEN
409:          is_feiertag:=true
410:        ELSE
411:          CASE ft OF
412:            hl_drei_koenige : is_feiertag:=land IN
413:                              [baden_wuertemberg,
414:                               bayern];
415:            fronleichnam    : is_feiertag:=land IN
416:                              [baden_wuertemberg,
417:                               bayern,
418:                               hessen,
419:                               nordrhein_westfalen,
420:                               rheinland_pfalz];
421:            mariae_himmelfahrt : is_feiertag:= land
                                                  IN
422:                                 [bayern,
423:                                  saarland];
424:            allerheiligen   : is_feiertag:= land IN
425:                              [baden_wuertemberg,
426:                               bayern,
427:                               nordrhein_westfalen,
428:                               rheinland_pfalz,
429:                               saarland];
430:          END;
431:      END;
432:
433:    BEGIN
434:      WITH feld DO
435:        BEGIN
436:          nr:=0;
437:          FOR i:=neujahr TO allerheiligen DO
438:            IF is_feiertag(i) THEN
439:              BEGIN
440:                nr:=succ(nr);
441:
datum_ft_berechnen(i,jahr,a[nr].tag,
                                          a[nr].monat,
442:                                      a[nr].bez);
443:              END;
444:          anz:=nr;
445:          quicksort(1,anz);
446:        END;
447:    END;
448:
449:  PROCEDURE berechne_matrix(     monat   ,
450:                                 jahr    : integer;
451:                             VAR feld    : ft_feld;
452:                             VAR matrix  : monats_
                                              matrix);
453:
454:    VAR i            ,
455:        j            ,
456:        t_index      ,
457:        wochentag : integer;
458:
459:    FUNCTION is_feiertag(tag    ,
460:                         monat : integer): boolean;
461:
462:      VAR i  : integer;
463:          ok : boolean;
464:
465:      BEGIN
466:        ok:=false;
467:        WITH feld DO
468:          FOR i:=1 TO anz DO
469:            ok:=ok OR ((tag=a[i].tag)
                       AND (monat=a[i].monat));
470:        is_feiertag:=ok;
471:      END;
472:
473:    BEGIN
474:      FOR i:=1 TO 6 DO
475:        FOR j:=1 TO 7 DO
476:          BEGIN
477:            matrix[i,j].tag:=0;
478:            matrix[i,j].fett:=false;
479:          END;
480:      i:=1;
481:      FOR t_index:=1 TO anzahl_tage(monat,jahr) DO
482:        BEGIN
483:          wochentag:=wochentag_berechnen(t_index,
                                          monat,jahr);
484:          IF wochentag=0 THEN
485:            BEGIN
486:              j:=7;
487:              matrix[i,j].fett:=true;
488:            END
489:          ELSE
490:            BEGIN
491:              j:=wochentag;
492:            matrix[i,j].fett:=is_feiertag(t_index,
                                              monat);
493:            END;
494:          matrix[i,j].tag:=t_index;
495:          IF j=7 THEN
496:            i:=succ(i);
497:        END;
498:    END;
```

Dem Floppy-Controller Dampf gemacht

1.2 MB auf einer Standarddiskette

In diesem Artikel wird von einer bisher beim ATARI ST nicht genutzten Möglichkeit die Rede sein, wie man noch mehr Daten auf seinen Standardisketten unterbringen kann. Bekannt ist bislang nur das sogenannte Höherformatieren, bei dem bis zu 11 Sektoren statt der standardmäßigen 9 Sektoren in eine Spur geschrieben werden. Auch die Anzahl der Spuren läßt sich im Rahmen der Laufwerksmöglichkeiten bis zu 87 pro Seite erhöhen (normal sind 79 Spuren). Eine derartig hochgepeppelte Diskette vermag ca. 950 kB Nutzdaten zu speichern und wird vom Betriebssystem des ATARI ohne äußere Eingriffe klaglos verwaltet.

Grundlagen

Die Diskette rotiert mit 300 Umdrehungen pro Minute, also 5 Umdrehungen pro Sekunde. Beim Formatieren der Diskette orientiert sich der Floppy-Controller (im weiteren Textverlauf FDC) an den Indeximpulsen. Bei jeder Umdrehung der Diskette wird ein solcher Impuls an den FDC geschickt, um ihm damit zu sagen: Hier fängt die Spur an bzw. hier ist die Spur zu Ende. Da mit der Standarddatenrate von 250 kBit/Sekunde gelesen und geschrieben wird und dem FDC für eine komplette Spur genau 1/5 Sekunde Zeit bleibt, können also maximal 6250 Bytes in eine Spur formatiert werden. Leider kann man aber nicht alle 6250 Bytes als Datenbytes nutzen, da jeder Track noch in Sektoren von je 512 Bytes unterteilt ist. Um diese Sektoren aber finden zu können, müssen sie noch mit bestimmten Bytefolgen gekennzeichnet werden. Zu jedem Sektor gehören noch etwa 50 Bytes an Steuerinformation. Das macht also pro Sektor 562 Bytes. Teilt man die maximale Anzahl der Bytes pro Spur durch eben diese Zahl 562, so kommt man auf einen Wert von knapp über 11; soviele Sektoren haben also genügend Platz in einer Spur. "Wie bekomme ich denn noch mehr Bytes in eine Spur?" werden Sie jetzt fragen. Einige helle Köpfe haben einfach die Geschwindigkeit Ihrer Floppy-Laufwerke gedrosselt. Dadurch kommen die Index-Impulse nicht mehr alle 200 Millisekunden, sondern - je nach Drehzahl - zB. alle 250 Millisekunden. Der FDC hat nun mehr Zeit, seine Daten in die Spur zu schreiben; es finden mehr als 11 Sektoren in der Spur Platz.

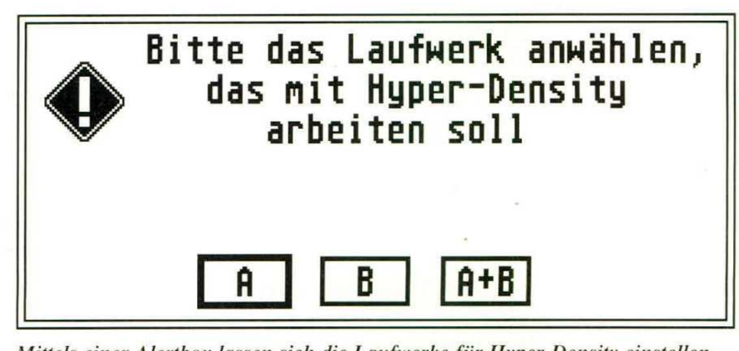
Kehrseite der Medaille: Die Geschwindigkeitseinstellung an der Floppy ist bei neueren Laufwerken kaum noch möglich, da diese quartz-gesteuert sind; ältere Laufwerke haben ein kleines Potentiometer, an dem die Drehzahl verstellt werden kann. Wie will man aber immer die richtige Drehzahl einstellen, und vor allem: Soll man das Besitzern nur einer Floppy wirklich zumuten?

Dem FDC an den Kragen

Wie wäre es denn, wenn man einfach die Datenrate des FDC erhöhte und so die Daten schneller als normal auf die Diskette schriebe? Genau von diesem Prinzip wird in diesem Beitrag Gebrauch gemacht. Der FDC wird mit einem Takt von 8 MHz versorgt und leitet davon alle Geschwindigkeiten ab: die Daten- (Lesen/Schreiben) und die Steprate. Die Steprate bezeichnet hierbei die Zeit, die dem Laufwerk zwischen den Kopfpositionierungen gegönnt wird.

Klemmt man den FDC von seinen 8 MHz ab und versorgt ihn mit einem Takt von 10.24 MHz, dann vollbringt er wahre Wundertaten. Warum aber nun gerade dieser krumme Takt von 10.24 MHz, werden Sie fragen. Rechnen Sie doch mal nach: bei 8 MHz Takt ist die Datenrate 250 kBit/Sekunde, also ist sie bei 10.24 MHz .... richtig, 320 kBbit/Sekunde. Dann passen statt der normalen 6250 Bytes satte 8000 Bytes in eine Spur, und das bei jeder normalen Floppy, egal ob 3.5 oder 5.25 Zoll. Rechnen wir mal weiter:

8000/562 = 14.23



Mittels einer Alertbox lassen sich die Laufwerke für Hyper-Density einstellen

Der Platz auf einer Spur reicht nun für 14 Sektoren!! Würde man einen Takt von genau 10 MHz benutzen, so fänden leider nur 13 Sektoren Platz in der Spur, außerdem ist der Wert von 10.24 MHz

schon fast Standard; teilen Sie mal fortlaufend durch 2: 10.24; 5.12; 2.56..... dämmert's?

Nachdem ich den FDC mit seinem neuen Takt von 10.24 MHz versorgt hatte, konnte ich schon über das Desktop oder andere Formatierprogramme die Praxistauglichkeit meiner Idee testen: Alles funktionierte wie erwartet. Nächster Schritt war die Modifikation des wohl hinlänglich bekannten Formatierprogramms Hyperformat von Claus Brod (Hallo Claus!) dahingehend, daß nunmehr 14 Sektoren pro Track zu formatieren waren. Mit Erfolg: 1162 kB auf einer zweiseitigen Diskette mit 83 Spuren. Die Umschaltung nahm ich in der ersten Zeit noch mit einem kleinen Schalter vor, was allerdings einige Einschränkungen hatte: Man konnte auf eine so formatierte Diskette nur von Festplatte oder von RAM-Disk kopieren oder aber von einer ebenso 'Hyper-Density'-formatierten Diskette, weil das Betriebssystem ja nicht umschalten konnte. Es mußte also ein Treiber geschrieben werden, der einerseits die Wahl des Laufwerks zuließ, welches mit Hyper-Density betrieben werden sollte, andererseits aber auch eine Anpassung der Stepraten zuließ, denn auch diese wurden durch den Eingriff kleiner, für manche Laufwerke eben zu schnell. Ein gewisses Maß an Bedienungsfreundlichkeit sollte auch gegeben sein. Untenstehendes Programm erfüllt, wie ich meine, all diese Bedingungen.

Es wird als Accessory beim Booten des Systems gestartet und fragt beim Anwählen nach, ob Sie mit Hyper-Density arbeiten wollen. Wenn ja, wählen Sie das oder die beiden Laufwerke aus und geben anschließend noch die Steprate ein, mit der das angewählte HD-Laufwerk positionieren soll. Erfahrungsgemäß sollten 3.5"- und 5.25"-Laufwerke mit 6 ms betrieben werden, Laufwerke mit Doppelstep sogar mit 12 ms. (Das sind jene Laufwerke, die zwar 80 Spuren haben, aber durch double-stepping auch 40-Spur MS-DOS-Disketten verarbeiten können.) Sinnvoll wird der Treiber natürlich erst, wenn Sie Ihren ST gemäß untenstehender Bauanleitung modifiziert haben. Natürlich kann im Accessory die Betriebsart Hyper-Density jederzeit wieder abgeschaltet werden, dann werden die Stepraten auch automatisch auf ihren Standardwert von 3 ms zurückgesetzt.



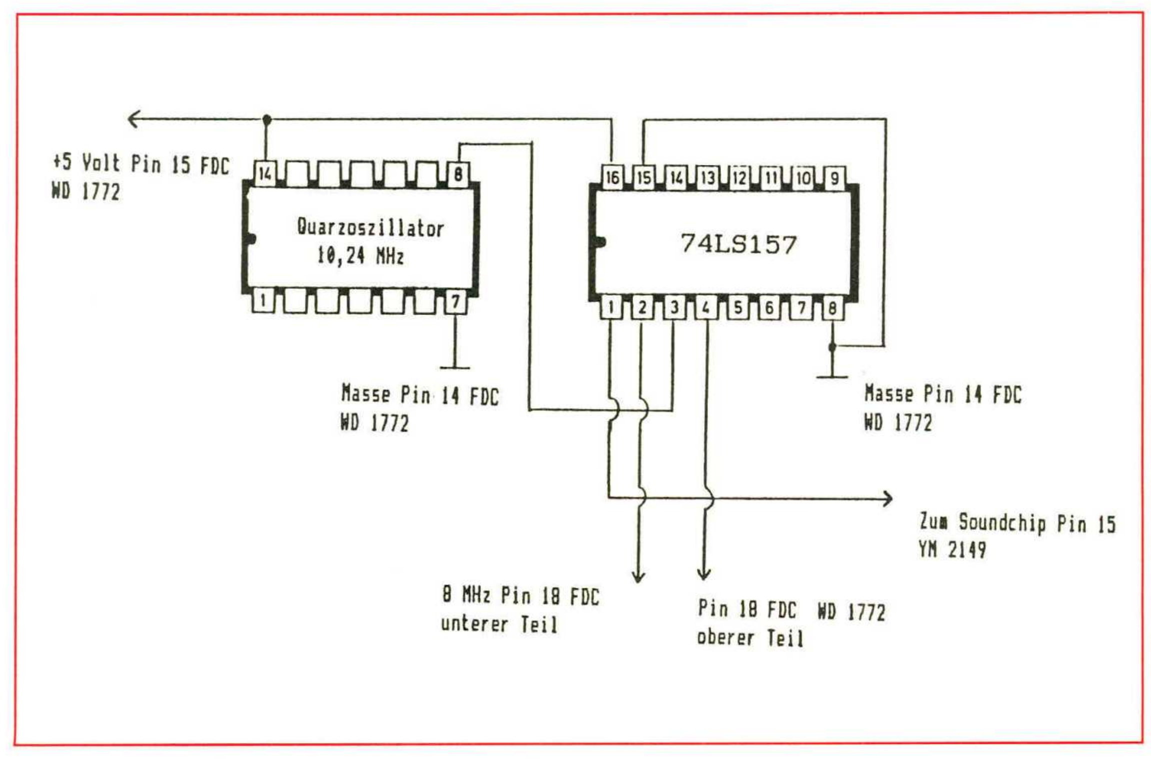
Einstellung der Steprate

Funktionsweise der Schaltung

Im wesentlichen besteht die Schaltung aus einem Quarzoszillator der Frequenz 10.24 MHz und einem IC vom Typ 74LS157. Aufgabe des ICs ist die Auswertung des Bit 6 vom Parallelport A des Soundchips. Ist dieses Bit auf 0 gesetzt, wie es im Einschaltzustand des ST der Fall ist, so wird der FDC mit dem normalen Takt von 8 MHz versorgt und arbeitet wie bisher. Sobald der Treiber aber erkennt, daß ein Laufwerk mit Hyperdensity angesprochen ist, setzt er Bit 6 des Soundchips. Das IC 74LS157 schaltet dann nicht mehr die 8 MHz, sondern die vom Oszillator gelieferten 10.24 MHz auf den FDC, welcher sodann auf die anders formatierten Disketten zugreifen kann.

Funktionsweise des Treiberprogramms

Das Programm installiert 4 neue Vektoren, von denen 3 beim Zugriff auf Massenspeicher benutzt werden, außerdem eine VBL-Routine, die bei jedem Vertical Blank Interrupt durchlaufen wird. Deren Aufgabe ist, festzustellen, ob ein vorher als Hyperdensity deklariertes Floppylaufwerk angesprochen wird, und dementsprechend Bit 6 im Port A des Soundchips zu setzen. Das Programm läuft unter allen TOS-Versionen und sollte als ACC gebootet werden. Ein Umbenennen in PRG und Start durch Doppelklick ist auch möglich, jedoch findet keine Prüfung statt, ob es schon einmal installiert wurde. Es verankert sich daher jedesmal aufs neue im Programmspeicher. Beim Ausschalten von Hyperdensity werden die Vektoren nicht deinstalliert, sondern lediglich deaktiviert, was einerseits die Struktur des Programmes vereinfacht, anderseits verhindert, daß andere vektorverbiegende Programme mit ausgeklinkt werden. Außerdem wer-



Schaltplan für Hyper-Density

den die veränderten Stepraten wieder auf ihren Standardwert von 3 ms zurückgesetzt. Der Treiber ist recht kurz gehalten und paßt daher gerade in zwei Sektoren (ein Cluster). Erreicht wurde dies durch konsequente Ausnutzung aller Codeoptimierungen, die nur wenige Assembler bereitstellen. Der Source stammt von einem GFA-Assembler; andere Assembler werden ein etwas längeres Programm generieren, was mit Belegen von 4 statt sonst 2 Sektoren 'belohnt' wird. Um das zu verhindern, können Sie auch den in GFA-BASIC geschriebenen Programmgenerator eintippen. Er generiert das fertige ACC mit einer Länge von 1021 Bytes.

Aufbauhinweise

Besorgen Sie sich das IC 74LS157 und einen Quarzoszillator der Frequenz 10.24 MHz, dazu einen 14poligen und einen 16poligen IC-Sockel. Der Oszillator hat ein völlig metallgekapseltes Gehäuse mit nur 4 Anschluß-drähtchen, genau passend für einen 14poligen IC-Sockel (alles erhältlich z.B. beim Völkner Elektronik-Versand). Bauen Sie gemäß unten abgebildeter Skizze die Schaltung auf. Dabei gilt bei allen angesprochenen ICs folgende Zählweise: Jeweils eine Schmalseite des ICs hat eine Einbuchtung oder einen Punkt. Der Pin, der entgegen dem Uhrzeigersinn an dieser Schmalseite liegt, ist immer Pin 1. Die anderen Pins sind dann weiter gegen den Uhrzeigersinn hochnumeriert. Schalten Sie unbedingt Ihren ATARI aus!! Am Floppy-Controller (28 Pin-Gehäuse WD 1772) trennen Sie den Pin 18 auf halber Höhe auf und verbinden die beiden Enden vorsichtig laut Schaltplan. Dann vom Soundchip (40-Pin Gehäuse YM 2149) Pin 15 verbinden mit Pin 1 des IC 74LS157. Zum Schluß stellen Sie die Masse- (Pin 14 YM 2149) und +5 Volt-Verbindung (Pin 15 YM 2149) her. Erst jetzt werden die beiden ICs in ihre Sockel gesteckt, aber bitte richtig herum! Checken Sie alles noch einmal durch, dann können Sie den ersten Probelauf starten. Alle normalen ATARI-Disketten sollten nun zu lesen und zu beschreiben sein. Erst wenn bis hierhin alles funktioniert, können Sie den Treiber booten und z.B. Laufwerk A als Hyperdensity deklarieren, Steprate 6 ms. Lassen Sie die Standarddiskette im Laufwerk A und versuchen Sie, ein Programm zu starten. Dieser Versuch sollte mit einem Fehler enden; schalten Sie mit dem Treiber-ACC Hyperdensity wieder aus: Jetzt sollten die Programme alle ohne Fehlermeldung zu starten sein.

```
 1:  ' Programmgenerator  (c) MAXON Computer GmbH
 2:  A$=Space$(1024)
 3:  Adr%=Varptr(A$)
 4:  Restore Daten
 5:  Pruef%=0
 6:  For N%=0 To 1023 Step 4
 7:    Read B$
 8:    Wert%=Val("&H"+B$)
 9:    Lpoke Adr%+N%,Wert%
10:    Pruef%=Pruef% Xor (Wert%)
11:  Next N%
12:  If Pruef%<>1540341029
13:    Alert 1," Daten nicht korrekt | Bitte überprüfen",1,"OK",Dummy%
14:  Else
15:    Bsave "HYDSEL.ACC",Adr%,1021
16:    Alert 1," Alles Ok ",1,"Ende",Dummy%
17:  Endif
18:  Edit
19:  '
20:  Daten:
21:  Data 601A0000,2A20000,12C0000,5280000,0,0,0
22:  Data 4BFAFEFE,4FFA08EC,4AAD0024,67124267,2F3C0000,9F66100,B43F3C
23:  Data 314E41,616E3AFC,A7001,2AC04295,61563D7A,53AFFE8,3AFC004D
24:  Data 70052AC0,42956144,3D7A0528,FFE63AFC,232AFC,10001,2ABC0001
25:  Data 3D7A,3840036,2D7C0000,2A2022E,611E3D7A,502FFEA,612A0C6E
26:  Data 28FFEC,66F6303A,370B07A,36266EC,614A60E8,223C0000,3B4303C
27:  Data C84E42,4DFA0362,2A4E4E75,3AFC0017,70012AC0,2AFC0001,2D7C
28:  Data 3E0,22E60D4,61DE2D48,22E3D40,363AFC,342AFC,10001
29:  Data 2ABC0001,60E2,61C251EE,FFDA41FA,1DE7001,61D60C6E,20176
30:  Data 662441FA,2267001,61C63D7A,47EFFDE,536EFFDE,50EEFFDA,41FA0268
31:  Data 700261B0,3D7A0468,FFDC42A7,3F3C0020,4E415C8F,2F0040E7,7C0700
32:  Data 4A2EFFDA,6604616C,600E6178,6100012E,4A2EFFDB,6602610C,46DF3F3C
33:  Data 204E41,5C8F4E75,41F80472,2D50FE3C,20BC0000,2345888,2D50FE26
34:  Data 20BC0000,21E5088,2D50FE52,20BC0000,24A3038,454E548,20780456
35:  Data 42814AB0,1000660C,45F01800,24BC0000,1D26006,5881B240,66E850EE
36:  Data FFDB4E75,612C610C,4A2EFFDA,670451EE,FFDA4E75,2F3A0232,3D7C0001
37:  Data FFDC3D7C,2FFDE,610000A6,2D5FFFDC,4E757240,60027200,41F88800
38:  Data 10BC000E,10100200,BF8001,11400002,4E75207A,2023010,B07A01F2
39:  Data 66024E75,48E74040,43FA020E,4A29FFDA,671E0C40,26416,3340FFD8
40:  Data C690002,FFDE6604,61B46008,B07A01CC,67F661AE,4CDF0202,4E755842
41:  Data 52414A53,48440000,2F3A,FFFA302F,1260B8,58425241,4A534844
42:  Data 0,2F3AFFFA,302F0008,60A25842,52414A53,48440000,2F3A
43:  Data FFFA302F,8608C,41F80A06,227804F2,C690100,26304,41F80A4C
44:  Data 43E8FFBC,2D49FFE2,50D143FA,13D303A,1581031,FF0C6E,2FFDE
45:  Data 6604610A,60120C6E,1FFDE,670A3140,231C0,4404E75,31400006
46:  Data 4E752020,48797065,722D5365,6C656374,20005B32,5D5B2020,20414343
47:  Data 2F505247,20BD3139,3839204A,53547C20,2020204D,94636874,656E2053
48:  Data 6965206D,69742020,20202020,7C204879,7065722D,44656E73,69747920
49:  Data 61726265,6974656E,203F205D,5B4E6569,6E7C4A61,5D005B31,5D5B4269
50:  Data 74746520,64617320,4C617566,7765726B,20616E77,84686C65,6E2C7C20
51:  Data 20206461,73206D69,74204879,7065722D,44656E73,6974797C,20202020
52:  Data 20206172,62656974,656E2073,6F6C6C5D,5B417C42,7C412B42,5D005B31
53:  Data 5D5B2020,42697474,65206469,65205374,65707261,74657C66,81722064
54:  Data 61732F64,69652048,442D4472,69766573,7C202020,20202061,6E676562
55:  Data 656E5D5B,33206D73,7C36206D,737C3132,206D735D,30001,3F4
56:  Data 40C,42A,56A,622,6C2,FFFF0000,5E2822
57:  Data 9E0C0C1C,1013E04,4040404,0
58:  Data *,1024 Bytes = 256 Data
```

Der Programmgenerator für Hyper-Density in GFA-BASIC

Jetzt geht's ans Formatieren: Leere Diskette in Laufwerk A, Hyperdensity für A einschalten und per Desktop formatieren. Jetzt haben Sie schon eine Diskette, die ein 'normaler' ST nicht mehr lesen kann, weil mit einer höheren Datenrate formatiert wurde. Was jetzt noch fehlt, ist das richtige Formatierprogramm, was ich Ihnen hier leider nicht präsentieren kann. Ich persönlich arbeite mit dem neuesten 'Hyperformat', was auch dieses Spezialformat unterstützt, leider aber nicht Public-Domain ist, sondern Teil der mitgelieferten Software zum Buch 'Scheibenkleister II' von Claus Brod und Anton Stepper, erschienen bei der MAXON Computer GmbH.

Insider-Tips

Laut Datenblatt des Herstellers verträgt der FDC Taktfrequenzen von bis zu 10 MHz. Versuche zeigten aber, daß selbst bei Frequenzen von 12 MHz noch keine Schwierigkeiten beim FDC auftraten. Eher versagen die Floppys ihren Dienst. Interessant wäre aber der Anschluß von 5.25"-HD-Laufwerken, welche eine Nutzdatenkapazität von 1.2 MB aufweisen und mit einem Format von 15 Sektoren pro Spur arbeiten. Allerdings arbeiten diese Drives mit 360 Umdrehungen pro Minute, was bei der Wahl der Taktfrequenz auch berücksichtigt werden muß.

Ein Einlesen des gesamten Tracks per

FDC-Kommando Read Track wird immer zu dem Ergebnis von 7168 Bytes führen, weil der DMA-Baustein auf solche Datenmengen nicht vorbereitet zu sein scheint. Das Lesen der ID- und Sektordaten klappt jedoch reibungslos.

Aufruf an alle Programmierer: Bindet doch diese Option in Eure Formatierroutinen ein; alle Sektoranzahlen über 11 bis 14 pro Track schalten Bit 6 Port A auf logisch 1, und schon kann man das neue Hyperdensity Format erzeugen!

Eigene Erfahrungen

Ich persönlich arbeite mit Hyperdensity und 13 Sektoren pro Track. Dabei ist die Geschwindigkeit des Einladens gegenüber Standarddisketten um einiges höher. Alle 4 meiner Laufwerke vertragen das neue Format, wenn die Steprate entsprechend angepaßt wird (6 ms). Auch in puncto Zuverlässigkeit konnte ich bisher keine nachteiligen Beobachtungen machen, es steht jedoch fest, daß mit höher werdender Datendichte auch rein statistisch höhere Fehlerquoten verbunden sind. Dies fällt besonders bei alten, oft gebrauchten Disketten in den Tracks über 80 auf, es kommt bisweilen schon einmal zu einem Formatierfehler bei solchen Disketten. Sie sollten daher nicht die ganz billigen Exemplare benutzen und auch nicht die 3 Jahre alten Disks aus der Schublade holen und 'überbrennen'. Die besten Erfahrungen habe ich mit einseitigen 'YES-NAME' Disketten gemacht, besonders mit allen Marken mit einem 'S' und einem 'O' im Namen...

Jürgen Stessun

Literatur:

1. Scheibenkleister ISBN 3-927065-00-5, Merlin Computer GmbH

2. ST Intern ISBN 3-89011-119-X, Data Becker GmbH

3. ATARI ST Profibuch ISBN 3-88745-501-0, Sybex-Verlag GmbH

4. Pocket Guide ISBN 3-88078-026-9, Texas Instruments

5. Storage Management Products Handbook 1986, Western Digital Corporation

```
 1:  ;********* ACC/PRG zum Einstellen des/der
                 Hyper-Density-Laufwerke
 2:              (c) MAXON Computer GmbH
 3:  HDV_BPB         equ $472
 4:  SOUND           equ $ffff8800
 5:
 6:                  .TEXT
 7:  RESERVE:        lea    RESERVE-256,a5 ; Startadresse
                                         Basepage retten
 8:                  lea    STACK,sp   ; der neue Stack
 9:                  tst.l $24(a5)     ; Parent vorhanden?
10:                  beq RESACC        ; nein,also ein ACC
11:
12:                  clr.w -(sp)       ; Dummy
13:                  move.l #256+ZUEND-RESERVE,-(sp)
                                       ; Programmlänge
14:                  bsr MAIN          ; Hauptprogramm
                                         abfahren
15:                  move.w #$31,-(sp) ; Resident
                                         lassen
16:                  trap #1
17:  ;*****************************
18:  ; Der ACC-Teil
19:  ;*****************************
20:  RESACC:         bsr FILA6           ; Zeiger setzen
21:                  move.w #10,(a5)+    ; Control-Array
                                           füllen
22:                  moveq.l #1,d0
23:                  move.l d0,(a5)+
24:                  clr.l (a5)
25:                  bsr AES
26:                  move.w INTOUT,APPID-CONTRL(a6)
27:
28:                  move.w #77,(a5)+    ; Graf Handle
29:                  moveq.l #5,d0
30:                  move.l d0,(a5)+
31:                  clr.l (a5)
32:                  bsr.s AES
33:                  move.w INTOUT,GRHANDLE-CONTRL(a6)
34:
35:                  move.w #35,(a5)+   ; Menu Register
36:                  move.l #$10001,(a5)+
37:                  move.l #$10000,(a5)
38:                  move.w APPID,INTIN-CONTRL(a6)
39:                  move.l #ACCNAME,ADDRIN-CONTRL(a6)
40:                  bsr.s AES
41:                  move.w INTOUT,ACCID-CONTRL(a6)
42:
43:  LOOP:           bsr.s MEVENT
44:                  cmpi.w #40,MSGBUFF-CONTRL(a6)
45:                  bne.s LOOP
46:                  move.w MSGBUFF+8(pc),d0
47:                  cmp.w ACCID,d0
48:                  bne.s LOOP
49:                  bsr.s MAIN
50:                  bra.s LOOP
51:
52:  AES:            move.l #AESPB,d1
53:                  move.w #$c8,d0
54:                  trap #2
55:  FILA6:          lea    CONTRL(pc),a6    ; A5 und A6
                          für indirekte Adressierung
56:                  movea.l a6,a5
57:                  rts
58:
59:  MEVENT:         move.w #23,(a5)+   ; Evnt Keyboard
60:                  moveq.l #1,d0
61:                  move.l d0,(a5)+
62:                  move.l #$10000,(a5)+
63:                  move.l #MSGBUFF,ADDRIN-CONTRL(a6)
64:  AESHELP:        bra.s AES
65:
66:  ALERT:          bsr.s FILA6
67:                  move.l a0,ADDRIN-CONTRL(a6)
                                             ; der Text
68:                  move.w d0,INTIN-CONTRL(a6)
                                    ; Default-Button
69:                  move.w #52,(a5)+  ; Form Alert
70:                  move.l #$10001,(a5)+
71:                  move.l #$10000,(a5)
72:                  bra.s AESHELP
73:  ;*****************************
74:  ; Beginn des Hauptprogramms
75:  ;*****************************
76:  MAIN:           bsr.s FILA6       ; A5 und A6 setzen
77:                  sf WAHL-CONTRL(a6) ; Wahl noch
                                         ungültig(0)
78:                  lea    TEXT1,a0   ; Hd benutzen?
79:                  moveq.l #1,d0     ; Default-Button
                                       = 1  (Nein)
80:                  bsr.s ALERT   ; Alarmbox darstellen
81:
82:                  cmpi.w #2,INTOUT-CONTRL(a6) ; ist
                                                  es HD?
83:                  bne AUSWERTE    ; nein
84:
85:  SETHD:          lea    TEXT2,a0 ; Welches Laufwerk?
86:                  moveq.l #1,d0   ; Default-Button
                                       1 = A
87:                  bsr.s ALERT     ; Box darstellen
88:                  move.w INTOUT,HDDRIVE-CONTRL(a6)
                          ; gew. Laufwerk(1,2,3)
89:                  subq.w #1,HDDRIVE-CONTRL(a6)
                     ; ergibt: A=0  B=1  A+B=2       →
```

Artikel	Preis
Atari Mega ST 1 + SM 124	1.649,-
Atari Mega ST 2 + SM 124	2.198,-
Atari Mega ST 4 + SM 124	3.398,-
Atari Laserdrucker SLM804	3.498,-
Monitor SM 124	398,-
Color Monitor SC 1224	798,-
Atari Megafile 30	998,-
Atari Megafile 60	1.798,-
VORTEX FESTPLATTEN	
HD 20 plus	999,-
HD 30 plus	1.098,-
HD 60 plus	1.798,-
3,5" Diskettenlaufwerk	298,-
5,25" Diskettenlaufwerk	398,-
EPSON-Drucker	
LX-800	479,-
EBZ LX-800 LQ 400/500	198,-
FX-850	1.149,-
FX-1050	1.449,-
LQ-500 24 Nadel A4	749,-
LQ-550 24 Nadel A4	849,-
LQ-850 24 Nadel A4	1.549,-
LQ-1050 dito A3	1.998,-
LQ-2550 dito A3	3.298,-
NEC P 2200 24 Nadel A4	879,-
NEC P6+ 24 Nadel A4	1.698,-
NEC P7+ 24 Nadel A3	1.998,-
Einzelblatteinzug P 6 +	449,-
Farb-Option P 6 + / P 7 +	259,-
Star LC-10	479,-
Star LC-10 color	698,-
Star LC 24-10	879,-
EBZ LC 10 / 24	229,-
Druckerkabel Atari	25,-
Druckerpapier 1000 Blatt	19,-
SOFTWARE Atari ST	
Beckerbase ST	89,-
Beckertext ST 2.0	179,-
Beckertools ST	89,-
Textomat 3.0 ST	89,-
Datamat ST	89,-
Kalkumat ST	89,-
Datamat ST Anwendungen	89,-
Hausverwaltung	449,-
Beckertext ST 2.0	269,-
Beckercad ST	449,-
GFA-Basic 3.0 Interpreter	179,-
GFA-Basic 2.0 Compiler	89,-
GFA-Basic 2.0 Interpreter	89,-
GFA-Draft Plus	298,-
GFA-Draft	179,-
GFA-Objekt	179,-
ST-Kontor FiBu	498,-
dBMAN Datenbank für ST	379,-
ProText 2.1 für ST	148,-
Signum!Zwei	398,-
Stad	169,-
Megamax C	349,-
Modula 2	349,-
IMagic	449,-
Daily Mail	169,-
BTX Manager V3.0 inkl. Interf.	398,-
PC dito 3.96	198,-
Superbase Professional	349,-
Superbase 2	179,-
LDW Power-Calc	209,-
Faktura Integ. Businesspaket	179,-
Publishing Partner d	209,-
ST Paint plus	109,-
Midisoft Studio Mehrspur-Sequenzer	119,-
Specdtrum-Malprogramm	119,-
Mark Williams C Vers. 3.0 d	249,-
Cyber Paint 2.0	109,-
csd-Source Level Debugger	119,-
Devpac Assembler 2.0	118,-
HiSoft-Basic Compiler	149,-
Adimens-Prog für GFA-Basic	179,-
CADproject Professional d	539,-
Adimens-Prog für Pascal Plus	179,-
Cyber Control	89,-
Antic Cyber Studio CAD 3D 2.0	149,-
SAVED Utility 2.0	89,-
Twist-Multiswitcher	70,-
SCHUTZHAUBEN aus hochwertigem Kunstleder anthrazit.	
ATARI 1040/260/520	24,95
FLOPPY SF 314/354	22,95
MONITOR 124/125	27,95
MONITOR SC s1224	27,95
MEGA ST Tastatur	24,95
MEGA ST Set	49,95
Drucker EPSON	
FX86/800/850/LX86/LQ500/850	24,95
FX 1000/1050/LQ1050/2550	27,95
Drucker NEC	
P2200	24,95
P 6/7 – P 6/7 Plus	27,95
Drucker STAR	
NL10/LC10/10c/24-10	24,95
Drucker PANASONIC	
1080/91/1092/1592	24,95

Achtung nur Versand
Abholung nur nach Voranmeldung möglich.

Für die Schweiz liefern wir ab Lager Zürich

TORNADO-Computer
Wangenerstr. 99 · 7980 Ravensburg
Tel. 07 51/39 51 · FAX 07 51/39 53





! Der ST kann sprechen !

TELL ME.PRG

Einfache Sprachausgabe.

TELL ME kann Briefe vorlesen oder in Gfa PRG's eingebunden werden. Sprachgeschwindigkeit, Silbenpausen etc. können beeinflußt werden. Die einfache Sprachqualität kann ggf. durch eigene Phoneme mit Ihrer Stimme u. einem Soundsampler verbessert werden.

Preis 24.-DM, mit Gfa-Source 49.-DM*
Nötige Hardware: 1MB, SW, *doppelseitiges Laufwerk.

Try Soft TELL ME-Kunden, die noch 88.- statt 49.- DM gezahlt haben, können die TEXTVERARBEITUNG **TEDI.PRG** bis Okt. 89 für 15.-DM beziehen.
Orginaldisk (MULTYTRONIK) und Rechnung einsenden.

ACHTUNG: Es werden unerlaubte Kopien für 89.-DM ohne Try Soft Etikett vertrieben. Sollten Sie diese Version besitzen, schicken Sie uns bitte die Rechnung.

Versand 5.-DM, Bestellung: nur V-Scheck oder NN an:

Try Soft **Ingeborg von Tryller**
32oo Hildesheim, Steinbergstr.6, Telefon: 05121 22882
Mitglied im ATARI Anwenderclub Hannover/Hildesheim

PICTURE-DISK Grafiksammlug
1000 Grafiken DM 90.- 2400 Grafiken DM 190.-
GFA ACCESSORY-MAKER 49.-
Wandelt GFA-Basicprogramme in ACCESSORYS um !!!
ZEICHENSATZ-PROFI 2.0 119.-
Kann innerhalb von 10 Minuten einen kompletten Zeichensatz für Signum II. Dies schließt das Erzeugen von Editor- und Druckerzeichensatz ein. ZSP hat auch Funktionen wie: hohl, fett, schmal, schatten, drehen, Sinusbiegung, Freihandbiegung, Gem-Font als Sigum-Font speichern, konvertieren, editieren, HEADLINE erzeugen, usw. INFORMATIONEN anf.
ART-PROFI FÜR SIGNUM 79.-
wie ZSP, es werden jedoch fertige Fonts benötigt. Erzeugt aus einem Font beliebig viele neue Fonts !
CompTec Software, OT Kriwitz 12
3131 Lemgow Tel. 05883/1325 für die Schweiz :
DataTrade AG, Langstraße 94, CH-8021 Zürich

NEC-Diskstation 3,5" ★ anschlußfertig **239,-**
NEC 1037 A ● 1MB ● Mit VDE-Netzteil u. Anschlußkabel ● Graues Gehäuse ● Graue Frontbl. ● Ein-/Ausschalter ● Gegen Aufpreis: Buchse für 2. Station 29,90 ● Umschaltung für 3. Station 49,90
NEC 1037 A (Rohlaufwerk) ohne Zubehör 179,-

Tastatur-Interface Anschlußfertig **98,-**
Für IBM-kompatible Tastaturen (102 Tasten) zum Anschluß an den Midi-Port ● Platine incl. Treiber und Anschlußkabel

ST Tastatur-Umbausatz komplett **135,-**
Umbau der Atari-Tastatur ● Lieferumfang: schmalere Tasten ● Rechteckige Funktionstasten ● Farblich abgesetzt ● Druckfedern ● Anschlagsystem ● siehe Test ST 3/89 ● Nicht für MEGA ST ●

Harddisk-Interface incl. Software **89,-**
Zum Anschluß IBM kompatibler HD u. Controller am DMA-Port Sonderpreis: mit HD Optimizer nur 149,- (Handbuch auf Diskette)

Floppy-Expander 3 Laufwerke am ST **89,-**
Integr. Treiber für 5,25"-Laufwerke ● LED-Anzeigen ● Resetfreies Umschalten ● Auch für Mega ST u. 1040, bitte angeben!

Harddisk-Optimizer zur Reorganisation **99,-**
Volle Platten werden wieder schnell ● Steigerung der Zugriffsgeschwindigkeit ● Überprüfen aller Speichermedien ● Sortieren der Directories ● Reorganisation aller Dateien ● Löschen der Lost-Cluster ● Anzeigen der Bad-Cluster/Fat-Verpointerung ● Protokoll auf Drucker u. Diskette ● Demo-Disk u. Info 15,-

Platinenlayout-Programme
PCB Edit: Für 24 Nadel-Drucker/Plotter (HPGL) **169,-**
Ausdruck 1:1 u. 2:1 ● Auflösung 1/180" ● Verarbeitung von Platine ST- u. MKP Editor-Layouts ● Gleichz. Darstellung u. Bearbeitung beider Platinenseiten ● Microleiterb. ● Bestückungspläne Platinengröße: 20,3 x 24 cm ● Demo-Disk u. Info 15,-
Layout ST: Für 9- u. 24-Nadel-Drucker ● 1/80" **149,-**
Ausdruck 1:1 u. 2:1 ● Microleiterbahnen ● Lötstopmasken ● Auch für ungenormte Raster ● Platinengröße 20 x 20 cm ● Kompatibler Autorouter ab Mitte 89 erhältlich ● Demo-Disk u. Info 15,-

GAL Programmierer für ST u. Mega ST **199,-**
Programmieren, Auslesen, Vergleichen u. Löschen der Standart GAL's 16 V 8 u. 20 V 8 ● Incl. GEM-Software u. ausführl. Anleitung Verlängerungskabel 25,- ● Aufpreis für 2 Textoolsockel 50,-

Stecker/Buchsen: Romport-Buchse 15,- ● Romport-stecker (Platine) 19,90 ● Floppy-/Monitor-Stecker/Buchse je 6,90 ● Weitere Stecker und Kabel auf Anfrage

Sonstiges: Romtos/Blitter-Romtos je 99,- ● (Inzahlungnahmen) dto. mit Fastload je 109,- ● TOS 1.4 auf Anfrage ● TOS-Umschaltung 199,- ● Romport-Expander/Verlängerung, 229,- Drucker-Umschalter für 2 Drucker, incl. Anschlußkabel 79,-

Ab einem Bestellwert von DM 80,- ist der HD-Optimizer zum Sonderpreis von DM 69,- erhältlich (Anleitung auf Diskette)

Versandkosten: Bei Nachnahme 8,- ● Bei Vorkasse (Scheck) 6,- Demo-Disk 3,- ● Auslandsversandkosten 12,- (Nur Vorkasse)

Wischolek Computertechnik
Mesteroth 9 ★ Bottrop 2 ★ Nur Versand
☎ **(0 20 45) 8 16 38** Mo-Fr. 10-11.30 u. 15-18 Uhr

```
 90:                    st WAHL-CONTRL(a6) ; Wahl ist
                                              gültig(-1)
 91:
 92:                    lea   TEXT3,a0   ; Welche Steprate?
 93:                    moveq.l #2,d0    ; Default = 6ms
 94:                    bsr ALERT        ; Box zeichnen
 95:                    move.w INTOUT,RATE-CONTRL(a6)
                                         ; Rate 1,2 oder 3
 96:  ;****************************
 97:  ; Auswertung der Box-Buttons
 98:  ;****************************
 99:  AUSWERTE:         clr.l -(sp)
100:                    move.w #32,-(sp) ; auf Super-Modus
                                           schalten
101:                    trap #1          ; Gemdos
102:                    addq.l #6,sp     ; Stackkorrektur
103:
104:                    move.l d0,-(sp)  ; alten Stack
                                           merken
105:                    move.w sr,-(sp)  ; Status merken
106:                    ori.w #$700,sr   ; Interrupts aus
107:
108:                    tst.b WAHL-CONTRL(a6) ; Wahl
                                                gültig?
109:                    bne MAKEHD       ; ja
110:
111:                    bsr DEINSTALL ; nein, deaktivieren
112:                    bra SUPEREND  ; und raus
113:
114:  MAKEHD:           bsr OLDSTEP   ; Stepraten auf
                                        normal (3ms)
115:                    bsr MKSTEP    ; Stepraten
                                        einstellen
116:                    tst.b INSTALLED-CONTRL(a6) ; schon
                                                installiert?
117:                    bne SUPEREND  ; ja, dann sofort
                                        raus
118:                    bsr INSTALL   ; neue Vektoren
                                        installieren
119:
120:  SUPEREND:         move.w (sp)+,sr  ; Status holen
121:                    move.w #32,-(sp) ; und wieder in
                                           User-Modus
122:                    trap #1          ; Gemdos
123:                    addq.l #6,sp
124:                    rts              ; fertig
125:  ;****************************
126:  ; Neue Vektoren installieren
127:  ;****************************
128:  INSTALL:          lea   HDV_BPB,a0  ; Adresse des
                                          1. alten Vektors
129:                    move.l (a0),OLDBPB-CONTRL(a6)
                                          ; merken
130:                    move.l #NEWBPB,(a0) ; und neuen
                                          Vektor vorgeben
131:
132:                    addq.l #4,a0        ; das ist HDV_RW
133:                    move.l (a0),OLDRW-CONTRL(a6)
134:                    move.l #NEWRW,(a0)
135:
136:                    addq.l #8,a0  ; das ist HDV_MEDIACH
137:                    move.l (a0),OLDMED-CONTRL(a6)
138:                    move.l #NEWMED,(a0)
139:  ;**********************************************
140:  ; Für TOS 1.0 muP eine VBL-Routine installiert
      werden, die beim erstmaligen
141:  ; ansprechen der HD-Laufwerke den Soundchip
      umschaltet.
142:  ; erreicht bei neueren Versionen, daP auch im
      Desktop formatiert werden kann.
143:  ;**********************************************
144:                    move.w $454,d0  ; Anzahl der
                                          VBL-Routinen
145:                    lsl.w #2,d0     ; mal 4
146:                    movea.l $456,a0 ; Adresse der
                                          VBL-Routinen
147:                    clr.l d1
148:  VBLT:             tst.l 0(a0,d1.w)  ; schon belegt?
149:                    bne VBL1          ; ja
150:                    lea   0(a0,d1.l),a2  ; Adresse für
                                             VBL merken
151:                    move.l #VBLNEU,(a2)  ; neue Routine
                                             eintragen
152:                    bra VBLEND           ; und raus
153:  VBL1:             addq.l #4,d1      ; nächste Adresse
154:                    cmp.w d0,d1       ; alle durch?
155:                    bne VBLT          ; nein
156:  VBLEND:           st INSTALLED-CONTRL(a6) ; Signal:
                                           installiert(-1)
157:                    rts
158:  ;*****************************
159:  ; HD nicht ansprechbar machen
160:  ;*****************************
161:  DEINSTALL:        bsr ND      ; Umschalten auf Normal
162:                    bsr OLDSTEP ; alte Stepraten
                                      einstellen
163:                    tst.b WAHL-CONTRL(a6) ; noch aktiv?
164:                    beq ISOUT             ; nein
165:                    sf WAHL-CONTRL(a6)    ; Signal:
                                              kein HD(0)
166:  ISOUT:            rts
167:  ;*****************************
168:  ; alte Stepraten einstellen
169:  ;*****************************
170:  OLDSTEP:          move.l RATE,-(sp) ; HD-Drives und
                                            Rate retten
171:                    move.w #1,RATE-CONTRL(a6)
                                        ; Steprate auf 3ms
172:                    move.w #2,HDDRIVE-CONTRL(a6)
                                        ; Beide Laufwerke
173:                    bsr MKSTEP      ; einstellen
174:                    move.l (sp)+,RATE-CONTRL(a6)
                                ; HD-Drives/Rate herstellen
175:                    rts     ; fertig
176:  ;*************************************
177:  ; Hier erfolgt die Umschaltung (Bit 6)
178:  ;*************************************
179:  HD:               moveq.l #$40,d1   ; Bit 6=1 für Hd
180:                    bra.s DENSITY     ; weiter geht's
181:  ND:               moveq.l #0,d1     ; Bit 6=0 für Nd
182:  DENSITY:          lea   SOUND,a0    ; Adresse des
                                            Soundchips
183:                    move.b #14,(a0)   ; Port A anwählen
184:                    move.b (a0),d0    ; einlesen in D0
185:                    andi.b #$ff-$40,d0 ; Bit 6
                                             ausmaskieren
186:                    or.b d1,d0        ; mit neuer
                                            Density odern
187:                    move.b d0,2(a0)   ; zurück in
                                            Port A
188:                    rts               ; fertig
189:  ;******************************
190:  ; die VBL-Routine (für TOS 1.0)
191:  ;******************************
192:  VBLNEU:           movea.l CDEV,a0   ; Adresse Current
                                            Drive
193:                    move.w (a0),d0    ; Drive -> d0
194:                    cmp.w CHECKED,d0  ; Schon geprüft?
195:                    bne CHECK         ; nein, dann los
196:                    rts               ; VBL fertig
197:  ;*************************************
198:  ; Routine zur Selektion der HD-Drives
199:  ;*************************************
200:  CHECK:            movem.l d1/a1,-(sp) ; wird noch
                                              gebraucht
201:                    lea   CONTRL,a1   ; zum Adressieren
202:
203:                    tst.b WAHL-CONTRL(a1) ; HD
                                            gewünscht?
204:                    beq DRIVEOK    ; nein
205:
206:                    cmpi.w #2,d0   ; Drive > 1?
                                         (C,D,...)
207:                    bcc DRIVEND    ; ja, also auf ND
                                         schalten
208:
209:                    move.w d0,CHECKED-CONTRL(a1)
                               ; bearbeitetes Drive merken
210:                    cmpi.w #2,HDDRIVE-CONTRL(a1)
                               ; beide Laufwerke?
211:                    bne DRIV1      ; nein
212:  MKHD:             bsr HD         ; ja, HD einstellen
213:                    bra DRIVEOK    ; und raus
214:
215:  DRIV1:            cmp.w HDDRIVE,d0  ; gewünschtes
                                            Drive?
216:                    beq MKHD       ; ja, also HD
217:
218:  DRIVEND:          bsr ND         ; auf Normal stellen
219:  DRIVEOK:          movem.l (sp)+,d1/a1 ; Register
                                              zurück
220:                    rts                 ; fertig
221:  ;*********************************
222:  ; Hier folgen die 3 neuen Routinen
```

→

```
223:  ;**********************************
224:                .DC.b 'XBRA','JSHD'
225:  OLDRW:        .DC.l 0
226:  NEWRW:        move.l OLDRW,-(sp)  ; alte Routine
                                          auf Stack
227:                move.w 18(sp),d0    ; das momentane
                                          Laufwerk
228:                bra CHECK           ; über CHECK in
                                          alte Routine
229:  ;*********************************
230:                .DC.b 'XBRA','JSHD'
231:  OLDBPB:       .DC.l 0
232:  NEWBPB:       move.l OLDBPB,-(sp) ; alte Routine
                                          auf Stack
233:                move.w 8(sp),d0     ; das Laufwerk
234:                bra CHECK           ; über CHECK in
                                          alte Routine
235:  ;*********************************
236:                .DC.b 'XBRA','JSHD'
237:  OLDMED:       .DC.l 0
238:  NEWMED:       move.l OLDMED,-(sp) ; alte Routine
                                          auf Stack
239:                move.w 8(sp),d0     ; das Laufwerk
240:                bra CHECK           ; über CHECK in
                                          alte Routine
241:  ;*********************************
242:  ; ändern der Steprate für die
243:  ; gewählten HD-Laufwerke
244:  ;*********************************
245:  MKSTEP:       lea    $a06,a0  ; DSB0 für ROM/RAM-
                                    TOS vom 6.2.86
246:                movea.l $4f2,a1  ; sysbase
247:                cmpi.w #$0100,2(a1) ; altes TOS?
                                          (1.0)
248:                bls.s OLDTOS        ; ja
249:                lea    $a4c,a0      ; DSB0 für
                                      BlitterTOS/TOS1.4
250:  OLDTOS:       lea    -$44(a0),a1  ; Current Drive
251:
252:  MKWERT:       move.l a1,CDEV-CONTRL(a6)
                              ; dort steht das mom. Drive
253:                st (a1) ; ungültig machen
254:                lea    TIMES,a1  ; dort stehen die
                                   Werte der Stepraten
255:                move.w RATE,d0  ; gedrückter Button
256:                move.b -1(a1,d0.w),d0 ; Wert in D0
257:                cmpi.w #2,HDDRIVE-CONTRL(a6)
                                 ; beide Laufwerke?
258:                bne TESTB       ; nein, dann B testen
259:                bsr LWA         ; ja, A setzen
260:                bra LWB         ; und B setzen
261:
262:  TESTB:        cmpi.w #1,HDDRIVE-CONTRL(a6)
                                ; B-Laufwerk?
263:                beq LWB         ; ja
264:  LWA:          move.w d0,2(a0) ; Werte für
                                  A-Laufwerk poken
265:                move.w d0,$440  ; Steprate für A
266:                rts             ; fertig
267:  LWB:          move.w d0,6(a0) ; Wert für
                                  B-Laufwerk
268:                rts             ; fertig
```

```
269:  ;*************************
270:  ;Beginn des Datenbereichs
271:  ;*************************
272:                .DATA
273:  ACCNAME:      .DC.b '  Hyper-Select ',0
274:  TEXT1:        .DC.b '[2][   ACC/PRG  JST|'
275:                .DC.b '    Möchten Sie mit     |'
276:                .DC.b ' Hyper-Density arbeiten ?]'
277:                .DC.b '[Nein|Ja]',0
278:
279:  TEXT2:        .DC.b '[1][Bitte das Laufwerk
                          anwählen,|'
280:                .DC.b '   das mit Hyper-Density|'
281:                .DC.b '   arbeiten soll]'
282:                .DC.b '[A|B|A+B]',0
283:
284:  TEXT3:        .DC.b '[1][  Bitte die Steprate|'
285:                .DC.b 'für das/die HD-Drives|'
286:                .DC.b '      angeben]'
287:                .DC.b '[3 ms|6 ms|12 ms]',0
288:
289:  TIMES:        .DC.b 3,0,1   ; Werte für 3ms, 6ms,
                                    12ms
290:  AESPB:        .DC.l CONTRL,GLOBAL,INTIN,INTOUT,
                          ADDRIN,ADDROUT
291:  CHECKED:      .DC.w $ffff
292:
293:
294:                .BSS
295:                .EVEN
296:  WAHL:         .DS.b 1
297:  INSTALLED:    .DS.b 1
298:  RATE:         .DS.w 1   ; Reihenfolge so lassen!
299:  HDDRIVE:      .DS.w 1   ; Reihenfolge so lassen!
300:  VERSION:      .DS.w 1
301:  CDEV:         .DS.l 1
302:
303:  GRHANDLE:     .DS.w 1
304:  APPID:        .DS.w 1
305:  ACCID:        .DS.w 1
306:  MSGBUFF:      .DS.w 10
307:  CONTRL:
308:  OPCODE:       .DS.w 1
309:  SINTIN:       .DS.w 1
310:  SINTOUT:      .DS.w 1
311:  SADDRIN:      .DS.w 1
312:  SADDROUT:     .DS.w 1
313:                .DS.w 7
314:  GLOBAL:       .DS.w 15
315:  INTIN:        .DS.w 80
316:  PTSIN:        .DS.w 80
317:  INTOUT:       .DS.w 80
318:  PTSOUT:       .DS.w 12
319:  ADDRIN:       .DS.w 80
320:  ADDROUT:      .DS.w 80
321:                .DS.w 200
322:  STACK:        .DS.l 1
323:  ZUEND:
324:                .END
```

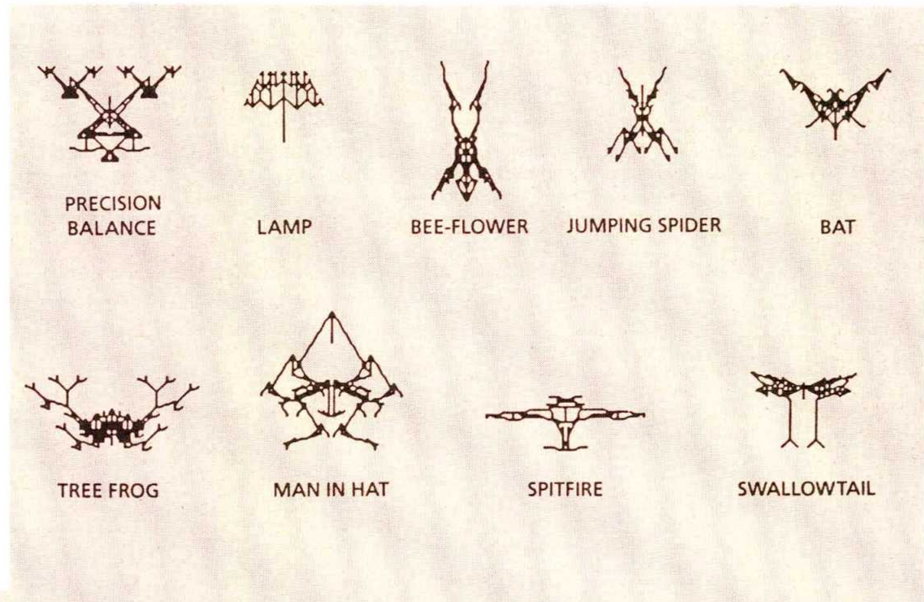
Das GFA-Assembler-Listing

Evolution im Computer

Vielleicht gewinnt die Frage, wohin uns unsere Vernunft führen wird, allein schon dadurch eine neue Bedeutung, daß wir mit Hilfe des Computers zu ergründen suchen, wie die Vernunft in die Welt gekommen ist.

Zweifellos halten wir viel vom logischen Denken, zählen es stolz zu einer unserer herausragenden intellektuellen Fähigkeiten. Dennoch müssen wir aus dieser Domäne einiges an den Computer abgeben, sind gezwungen, mitanzusehen, wie ein Automat gewisse mathematisch-logische Verknüpfungen in der Regel schneller bearbeitet als der Mensch und einige jener Problemstellungen zuweilen auch löst. Dagegen, über Jahrzehnte in den Hintergrund der Aufmerksamkeit und Wertschätzung verbannt, rückt nunmehr ein anderer Aspekt der Intelligenz, nämlich die Fähigkeit, Analogien zu ziehen, in den Blickwinkel des Interesses. Noch kann diese Komponente menschlichen Denkens als nicht 'computerisierbar' präsentiert werden und so im Vergleich zu den Maschinen verlorenes intellektuelles Terrain wiedergutmachen. Die wenigen existierenden Analogie-Programme (z. B. ANALOGY von Thomas G. Evans) halten einem Vergleich kaum stand, da hier keine wirklichen Analogieschlüsse stattfinden, sondern nur simple mechanische Verschiebungen vollzogen werden.

Allerdings, anders als erwartet, verhilft auch diese - bislang dem Menschen vorbehaltene - Fähigkeit dem Computer zu neuem Ansehen. Auf zweierlei Art: Zum einen wird der Rechner genutzt, um in einem quasi analogen Verfahren bestimmte natürliche Prozesse nachzuvollziehen, hierunter fallen u. a. Simulationsprogramme; zum zweiten dient die Maschine selbst als Analogieobjekt, beispielsweise als Modell zur Erklärung natürlicher Informationsspeicherung. Vom Computer als Analogiemaschine (wohlge-



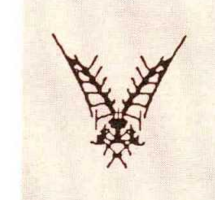
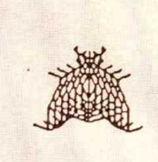
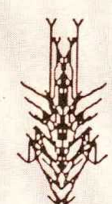
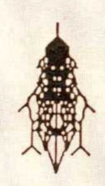
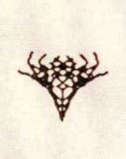
      

Bild 1: Biomorphe aus Dawkins "The Blind Watchmaker"

merkt, weder erkennt noch erzeugt der Computer irgendwelche Analogien, alles, was er tut, ist, in einem als vergleichbar postulierten Verfahren Ergebnisse analog natürlichen Ereignissen zu produzieren) und vom Computer als Analogieobjekt handelt dieser Artikel. Zugleich werden Informationen zur Evolutionstheorie mitgeliefert. Als Grundlage dient das meisterhaft geschriebene Buch "The Blind Watchmaker" von Richard Dawkins.

Dawkins, britischer Zoologe, Autor des Bestsellers "The Selfish Gene" (deutsch: "Das egoistische Gen"), Schüler des Nobelpreisträgers Niko Tinbergen, bietet in seinem 1986 zum ersten Mal erschienenen Buch "The Blind Watchmaker" (ein Bestseller)eine Fülle von Daten und Fakten innerhalb einer exzellenten Argumentation zugunsten der klassischen Evolutionslehre nach Charles Darwin dar. Selbst Kritiker, die sich nicht so leicht damit abfinden wollen, daß alles, wie es ist, ziellos und rein aus blindem Zufall entstand, die auf irgendeinen Sinn der Vernunft hoffen, können sich nur schwer, manchmal gar nicht der überzeugend aufgebauten Argumentationskette und den ungewöhnlichen Thesen und Beispielen entziehen.

Der blinde Uhrmacher

Machen wir einen kurzen Abstecher und lassen - wie Dawkins - William Paley, einen Theologen des 18. Jahrhunderts, zu Worte kommen. Um die Existenz eines Schöpfers, verantwortlich für die Komplexität und Schönheit des Lebens, zu 'beweisen', erfindet er folgende kleine Geschichte:

"Über die Heide wandernd, nehmen wir an, stoße ich mit meinem Fuß gegen einen Stein und werde gefragt, wie der Stein dorthin gekommen ist. Ich würde möglicherweise antworten ... er hat dort schon ewig gelegen ... Aber angenommen, ich hätte eine Uhr auf dem Boden gefunden, und es würde nachgefragt werden, wie die Uhr auf diesen Platz gelangt ist, würde ich kaum an die Antwort denken, die ich zuvor gegeben habe ... die Uhr muß einen Uhrmacher gehabt haben." (frei übersetzt)

Auf unsere Zeit übertragen: Glauben wir, daß es den Computer schon immer gegeben hat? Natürlich nicht. Er ist ein relativ junges Produkt des Menschen. Gleichwohl kennen wir bereits eine Evolutionsgeschichte des Computers, angefangen bei frühen, rein mechanischen Konstruktionen über Röhrenwerke, Transistorenvernetzungen, den integrierten Schaltungen bis schließlich zu den superdichten Schaltungen des letzten Stands der Technik. Den Menschen nun, hat es ihn von Urbeginn an gegeben? Nicht nur orthodoxe Darwinisten verneinen dies. Sie führen eine lange Abstammungsliste auf, beginnend bei der heutigen Menschheit über menschenähnliche und tierische Vorfahren bis hin zu den allerersten Formen des Lebens; sie zeichnen mit Akribie einen Baum des Lebens, dessen eine Wurzel sich irgendwo in der Urzeit der Erde vor etwa 3,5 - 4 Milliarden Jahren verliert. Die Neodarwinisten schließen einen seine Arbeit überdenkenden Uhrmacher, der mit weiser Voraussicht nach einem sinnvollen Plan all die wunderbaren Formen des Lebens erschaffen haben könnte, aus. Auch sie glauben (und das ist wörtlich zu nehmen, so finden wir ein 'Glaubensbekenntnis' Konrad Lorenz' an die Macht der Selektion am Schluß seines Buches "Das sogenannte Böse") an die natürliche Auslese, und diese hat per Definition keinen Verstand, keinen Plan, kein Ziel; sie ist blind, Gefährtin des Zufalls. Die natürliche Auslese, wird uns gesagt, ist die stetige Schöpferin allen Werdens - eine blinde Uhrmacherin.

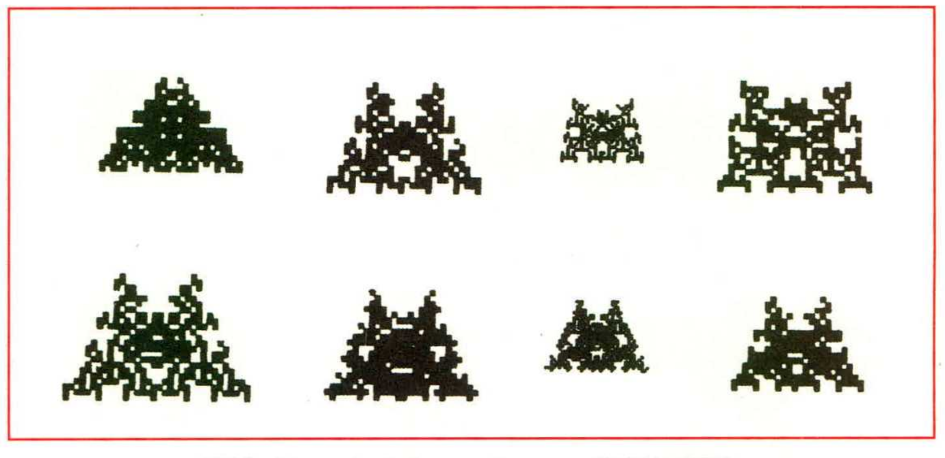


Bild 2 : Biomorphe, kreiert vom Programm UHRMACHER

Wer dies nicht einzusehen vermag, der sollte sich von Dawkins erzählen lassen, wie das Auge, ein hochkomplexes Gebilde, aus dem Nichts hat entstehen können. Nicht in einem großen Sprung, dazu hätte es ja eines intelligenten Schöpfers bedurft, nein, in unendlich vielen, winzigen Schritten, deren jeder sich im Druck der Außenwelt bewähren mußte, aufbauend auf den Ergebnissen aller Schritte, die vordem als erfolgreich ausgelesen wurden, bewertet nach den Prinzipien der kumulativen Selektion. Es ging, argumentiert Dawkins, beim Aufbau des Auges nicht um die alleinige Alternative zwischen den Extremen sehen und nicht sehen. Bereits die Unterscheidung von hell und dunkel - ein bißchen sehen also - brachte einen entsprechend kleinen Vorteil gegenüber der totalen Blindheit. Das gleiche Prinzip soll auf Flügel, Ohren, das Sonarsystem der Fledermäuse, die elektrischen Eigenschaften einiger Fische, die Giftzähne der Schlangen, das Verhalten der Ameisen und vieles mehr zutreffen.

Die Analogie von der DNS und dem ROM

"Es regnet DNS", schreibt Dawkins, und meint damit die Samen von Weiden oder Pusteblumen, die an ihren Fallschirmen im leichten Wind durch die Luft segeln. Ohne Ironie, so bemerkt der Autor, hätte er schreiben können: 'Es regnet Instruktionen, Programme, Algorithmen, ja, im übertragenen Sinne, es regnet Floppy-Disketten.' Warum? DNS - eine Abkürzung für das Riesenmolekül, auf dem die Erbinformationen gespeichert sind - wird verglichen, sogar gleichgesetzt mit dem Festwertspeicher eines Computers, dem ROM. DNS = ROM. Nur einmal, wie auch beim ROM, wird in der DNS die Gesamtinformation fest 'eingebrannt', und zwar in dem Augenblick, wenn ihre Bestandteile bei der Geburt der Zelle, in der sie sich befinden, neu zusammengestellt werden. Mit jeder Zellteilung der Mutterzelle teilt sich im folgenden auch die DNS, und ihre Informationen werden je nach Bedarf anschließend millionenfach ausgelesen, die dort gespeicherten Instruktionen und Programme steuern mit Hilfe von Enzymen (Eiweißen) den Aufbau des frisch gezeugten Lebewesens, ob Pusteblume oder Mensch.

Sowohl im Computer als auch in der DNS - eine weitere These Dawkins' -werden

Informationen in digitalen, d. h. qualitativ unterscheidbaren Werten abgelegt; in der Datenverarbeitung kennt man zwei Schaltzustände, die Natur verwendet in der DNS vier verschiedene Basen. Die Annahme einer digitalen Informationstechnik der Gene ist geradezu eine notwendige Voraussetzung für eine neodarwinistische Evolutionstheorie. Denn: Würden Informationen nicht in diskreten Einheiten gespeichert, erhielte man bei gemischter Vererbung gemischte Nachkommen, aus denen die ursprünglichen Eigenschaften nicht mehr rekonstruierbar wären. Zum Beispiel: Weißer Farbstoff vermengt mit schwarzem ergibt einen Grauton. Sooft auch Grau und Grau zusammengemischt werden, es ist nahezu ausgeschlossen, die ursprünglichen Farbanteile wieder zum Vorschein zu bringen. Seit den Versuchen Johann Gregor Mendels aber wissen wir, daß in der Nachkommenschaft zweier genetischer Mischformen einige Generationen später die reinen Ausgangstypen erneut auftreten werden.

Ein Computerlaie, der, jenseits des Schutzes der Dateiverwaltung seines Betriebssystems, mit einem sogenannten Diskettenmonitor direkt die Speicherwerte auf der Diskette unter die Lupe nimmt, wird erstaunt sein, in welcher scheinbaren Unordnung Datenstücke großer Dateien durcheinander zu liegen gekommen sind. Bald jedoch wird er erkennen, daß die Lage der Dateisektoren auf dem physikalischen Datenträger im Prinzip unwichtig ist, die geordnete Zusammengehörigkeit verschiedener Bruchstücke einer Datei wird im Diskettеninhaltsverzeichnis oder in speziellen Dateien vermerkt. So können ohne weiteres Teile bereits gelöschter oder überschriebener Dateien sich noch lange auf einer Diskette befinden. Eine Löschung muß nicht unbedingt bedeuten, daß alle Werte auf Null gesetzt werden, es reicht, einen Zeiger auf den Beginn der Datei zu löschen. Die solchermaßen getilgten Dateien werden vom Betriebssystem ignoriert, da im Inhaltsverzeichnis kein Zeiger diese Stellen der Diskette mit einem existierenden Dateinamen in Verbindung bringt, ihr Platz wird vom Betriebssystem zum Neubeschreiben freigegeben.

Im gleichen Sinne charakterisiert Dawkins auch die DNS, in der als 'Introns' bezeichnete Bruchstücke neue Bedeutung gewinnen. Diese Stellen im Text der DNS wurden bisher als sinn- oder inhaltslos eingestuft. In der neodarwinistischen, zugleich reduktionistischen Sicht [in welcher die Eigenschaften des Lebens vollständig auf die Gesetze der Physik zurückgeführt werden; die Existenz irgendeiner 'Lebensschwungkraft' oder 'élan vital' (Bergson) wird vollkommen ausgeschlossen] vermutet man in ihnen Reste alter Programme, die längst aus dem aktuellen Repertoire der DNS gestrichen worden sind.

Leben ist im Prinzip Information, so lautet eine der Hauptthesen Dawkins' (oder wie ein methodologischer Reduktionist, der Eigen-Schüler Bernd-Olaf Küppers, schreibt: "Leben = Materie + Information"). Eine Information, die wie im Computer digital abgespeichert wird.

Da hier nicht der Ort ist, wissenschaftliche Dispute auszutragen, empfehle ich allen am Thema Interessierten unbedingt abweichende (Hypo-) Thesen zu lesen, beispielsweise Arthur Koestlers "Der Mensch - Irrläufer der Evolution" oder die Aufsätze der beiden Physiknobelpreisträger Werner Heisenberg und Erwin Schrödinger in B.-O. Küppers

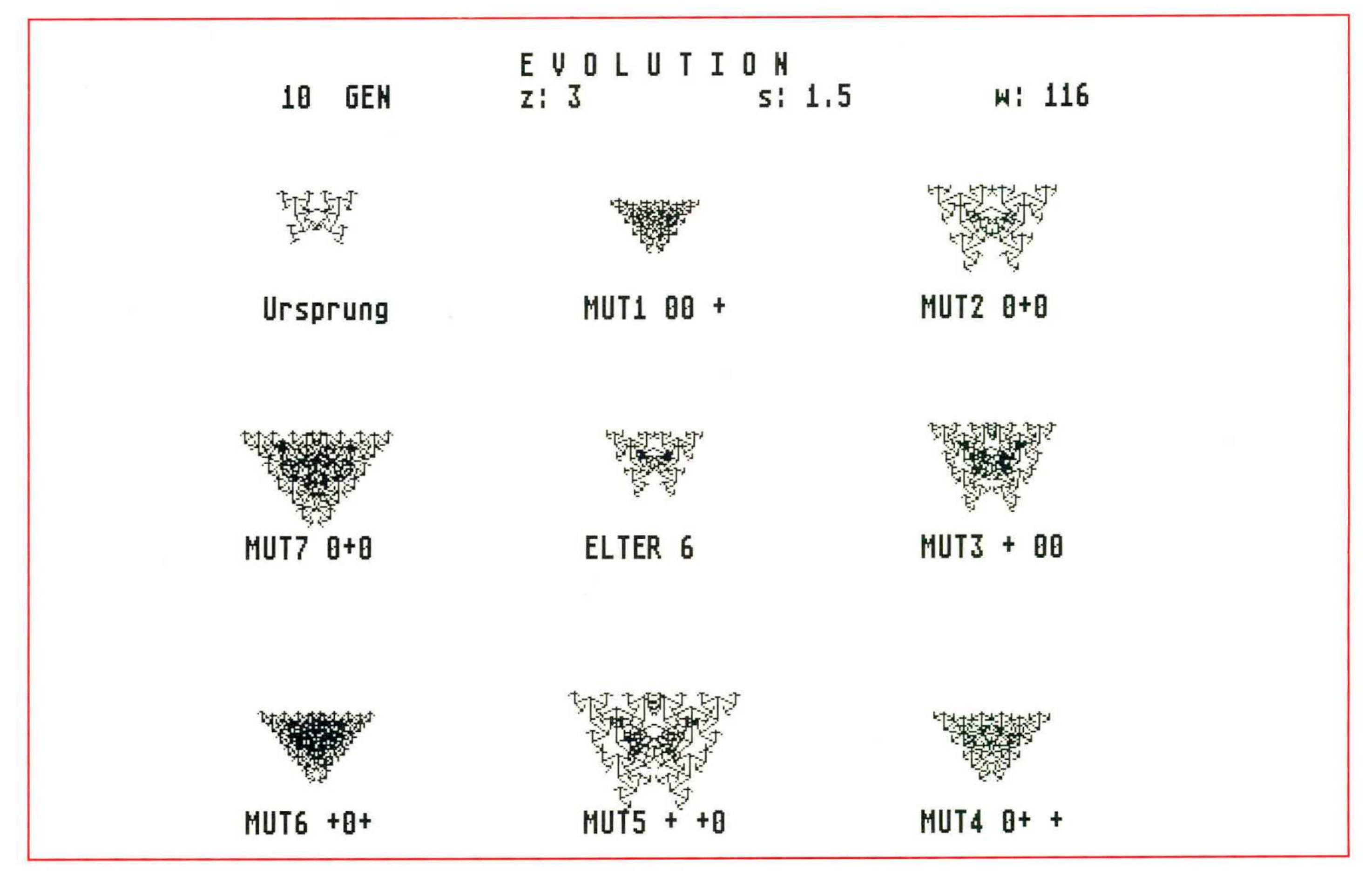


Bild 3: Bildschirmanzeige des Programms UHRMACHER

"Leben = Physik + Chemie?" oder "Die Natur des Vitalismus" vom Entdecker der DNS, Francis H. C. Crick oder "Evolution from Space" von Sir Fred Hoyle und Chandra Wickramasinghe oder die zahlreichen Ausführungen Karl R. Poppers zum Reduktionismus oder insbesondere "Das Geheimnis der Evolution" von Gordon Rattray Taylor.

Im Reigen der Biomorphe

Nach der Theorie die Praxis: Dafür setzen wir den Computer als Analogiemaschine ein und simulieren mit Hilfe eines kurzen, auf ein Minimum reduzierten und vereinfachten Programms die Grundzüge von Zellteilung, Mutationen und Vererbung im Prozeß der Evolution. Das Programm wird weniger komplex als Dawkins' Version sein, differenzierter aber als A. K. Dewdneys Vorschlag im Scientific American (1988/2). Die Resultate seines Programms nennt Dawkins Biomorphe, nach einer Bezeichnung von Desmond Morris (bekannt durch sein Buch "Der nackte Affe"). Biomorphe sind figürliche Gebilde, die Formen und Gestalten des Lebens verkörpern.

Was braucht man? Ein Programm, das Zellteilung, Wachstum, Mutationen und eine evolutive Entwicklung graphisch nachbildet. Auf die Simulation verschiedenartiger Geschlechter kann man verzichten, eine Darstellung ungeschlechtlicher Vermehrung, wie sie ja in der Natur häufig vorkommt, reicht aus.

Als Zellteilungsregel analog der Embryonalentwicklung, stark simplifiziert, hat Dawkins ein rekursives two-way branching-Verfahren gewählt. Das bedeutet: Zuerst wird eine vertikale Linie gezeichnet (entfällt in diesem Programm), die sich dann in zwei abknickende Geraden gabelt. Jede dieser beiden Verzweigungen erhält wiederum zwei Abzweigungen, die sich ebenfalls teilen usw. Die Entwicklung der Figur verläuft symmetrisch und ähnelt dem Wachstum von Bäumen.

Drei 'Gene' stehen zur Verfügung:

1. die Länge der zu zeichnenden Linien,
2. der Knickwinkel und
3. die Anzahl rekursiver Verzweigungen.

Die Mutation eines Gens wird dadurch symbolisiert, daß entweder Länge, Winkel oder die Anzahl der Verzweigungen um jeweils einen bestimmten Betrag ver-

```
1:  ' EVOLUTION v. Dr.A.Ebeling
2:  '
3:  ' (c) MAXON Computer GmbH
4:  DO
5:    evolution
6:  LOOP
7:  END
8:  '
9:  PROCEDURE evolution
10:   initialisieren
11:   ein=RANDOM(7)+1
12:   SELECT ein
13:   CASE 1
14:     w=w+1
15:   CASE 2
16:     s=s+s1
17:   CASE 3
18:     z=z+z1
19:   CASE 4
20:     s=s+s1
21:     w=w+w1
22:   CASE 5
23:     z=z+z1
24:     s=s+s1
25:   CASE 6
26:     z=z+z1
27:     w=w+w1
28:   CASE 7
29:     z=z+z1
30:     s=s+s1
31:     w=w+w1
32:   ENDSELECT
33:   gen=gen+1
34:   CLS
35:   PRINT AT(30,1);
      "E V O L U T I O N"
36:   PRINT AT(15,2);gen;"  GEN"
37:   PRINT AT(30,2);"z: ";z
38:   PRINT AT(45,2);"s: ";s
39:   PRINT AT(60,2);"w: ";w
40:   elter
41:   ursprung
42:   mut1
43:   mut2
```
→

```
44:   mut3
45:   mut4
46:   mut5
47:   mut6
48:   mut7
49:   WHILE INKEY$<>" "
50:   WEND
51: RETURN
52: '
53: PROCEDURE initialisieren
54:   PRINT "inittalisieren"
55:   xpos=420
56:   ypos=120
57:   ypos1=100
58:   zweige=2.5
59:   strecke=1.5
60:   winkel=105
61:   z=zweige
62:   s=strecke
63:   w=winkel
64:   z1=0.5
65:   s1=0.5
66:   w1=11
67:   dehn=3
68:   xd=50
69:   yd=40
70: RETURN
71: '
72: PROCEDURE ursprung
73:   x=xpos-ypos*2-xd
74:   y=ypos-yd
75:   ausgabe$=" Ursprung"
76:   reproduktion
      (zweige,strecke,winkel)
77: RETURN
78: '
79: PROCEDURE mut1
80:   ausgabe$="MUT1 00 +"
81:   x=xpos-ypos
82:   y=ypos-yd
83:   reproduktion(z,s,w+w1)
84: RETURN
85: '
86: PROCEDURE mut2
```
→

```
87:     ausgabe$="MUT2 0+0"
88:     x=xpos+xd
89:     y=ypos-yd
90:     reproduktion(z,s+s1,w)
91:   RETURN
92:   '
93:   PROCEDURE mut3
94:     ausgabe$="MUT3 + 00"
95:     x=xpos+xd
96:     y=ypos1*2
97:     reproduktion(z+z1,s,w)
98:   RETURN
99:   '
100:  PROCEDURE mut4
101:    ausgabe$="MUT4 0+ +"
102:    x=xpos+xd
103:    y=ypos1*3+yd
104:
reproduktion(z,s+s1,w+w1)
105:  RETURN
106:  '
107:  PROCEDURE mut5
108:    ausgabe$="MUT5 + +0"
109:    x=xpos-ypos
```

→

```
110:    y=ypos1*3+yd
111:
reproduktion(z+z1,s+s1,w)
112:  RETURN
113:  '
114:  PROCEDURE mut6
115:    ausgabe$="MUT6 +0+"
116:    x=xpos-ypos*2-xd
117:    y=ypos1*3+yd
118:
reproduktion(z+z1,s,w+w1)
119:  RETURN
120:  '
121:  PROCEDURE mut7
122:    ausgabe$="MUT7 0+0"
123:    x=xpos-ypos*2-xd
124:    y=ypos1*2
125:
reproduktion(z+z1,s+s1,w+w1)
126:  RETURN
127:  '
128:  PROCEDURE elter
129:    ausgabe$="ELTER
"+STR$(ein)
130:    x=xpos-ypos
```

→

```
131:    y=ypos1*2
132:    reproduktion(z,s,w)
133:  RETURN
134:  '
135:  PROCEDURE
reproduktion(z,s,w)
136:    DRAW "co 13"
137:    x1=x-35
138:    y1=y+35
139:    TEXT x1,y1+20,ausgabe$
140:    DRAW "co 6"
141:    DRAW "pu"
142:    SETDRAW x,y,0.35
143:    DRAW "pd"
144:    entwicklung(z,s,w)
145:  RETURN
146:  '
147:  PROCEDURE
entwicklung(zweige,strecke,winkel)
148:    IF zweige<>0 THEN
149:      DRAW "lt",winkel,"
fd",strecke*zweige*dehn
150:      entwicklung(zweige-
z1,strecke,winkel)
```

■

größert werden oder aber die Werte - sozusagen in einer Null-Mutation - gleichbleiben. Bei drei Genen und zwei Mutationszuständen (unverändert = <0>; vergrößert = <+>) sind sieben verschiedene Kombinationen möglich, in denen mindestens eines der drei Gene sich verändert, nämlich: <00+>, <0+0>, <+00>, <0++>, <++0>, <+0+> oder <+++>. Die Mutationsrate wird der Einfachheit halber auf ein Maximum gesetzt, d. h. in jeder Generation treten alle Mutationsformen auf.

Die Aufgabe der Selektion übernimmt der Anwender, indem er sich für eine der sieben Mutationen entscheidet, die dann zum Ausgangsobjekt (Elter) für die nächste Generation wird usw. Auch ein Zufallsgenerator könnte dazu eingesetzt werden. Dawkins malt sich ein Programm aus, das Blumenbilder farbig auf einen berührungsempfindlichen Bildschirm ausgibt, um mit seinen Farben Insekten anzulocken, die, wenn sie gegen den Bildschirm stoßen, durch die Berührungen selektiv die Evolution im Computer steuern; so entstünden auf natürlichem Weg biomorphe Blumen.

Wegen der leichten Programmierbarkeit graphischer Ausgaben und der problemlosen rekursiven Programmierung wurde der Algorithmus in der Sprache LOGO umgesetzt und anschließend auf GFA-BASIC angepaßt. Das Unterprogramm, in dem die Verzweigungsregel vorherrschend ist, nennen wir - nach Dawkins - ENTWICKLUNG. ENTWICKLUNG selbst ist dem Teilprogramm REPRODUKTION untergeordnet, das schließlich in das Hauptprogramm EVOLUTION eingebettet ist.

Als Dawkins sein Programm, das einen differenzierteren Aufbau als das hier vorgeschlagene aufweist (u. a. Einbezug einer größeren Anzahl von Genen und Mutationszuständen), zum ersten Mal laufen ließ, da erwartete er allenfalls Figurenvariationen nach dem Muster von Bäumen. Weder seine Intuition als Biologe noch die zwanzigjährige Erfahrung mit der Programmierung von Computern, schreibt er, hat ihn ahnen lassen, welche Ergebnisse er tatsächlich erhalten würde. Dann aber zeichneten sich Generation um Generation immer schärfer die Figuren von Insekten ab. "Deutlich hörte ich im Inneren die triumphalen Eröffnungsakkorde von 'Also sprach Zarathustra' (Filmmusik aus '2001')", berichtet Dawkins und fährt fort: "Ich konnte nicht essen, und als ich in jener Nacht versuchte einzuschlafen, schwärmten 'meine' Insekten hinter meinen Augenlidern."

Diese Vielfalt des Dawkinsschen Programms (s. Bild 1) kann unsere Version in der jetzigen Form nicht erzeugen, aber immerhin kommt es über die Ebene reiner Baumstrukturen hinaus (s. Bild 2 und 3). Man sollte mit dem Programm spielen, die Ausgangswerte verändern, neue Gene einbauen, ihre Mutationsmöglichkeiten vervielfachen etc. Der Phantasie sind

keine Grenzen gesetzt. Und unserer Analogiefähigkeit?

Zurück zum Anfang

Beschäftigen wir uns nun mit der Frage, wie Leben entstanden sein könnte. Wirklich nur aus dem Nicht-Leben in kleinsten Schritten, allein durch blinde Selektion weiterentwickelt? Auch darauf weiß Dawkins eine Antwort. Die 'Ursuppen-Theorie', nach der man nur die Bestandteile der Ur-Atmosphäre, also Wasserstoff, Wasserdampf, Ammoniak, Kohlendioxyd und Methan zusammenmischen und - Gewitter simulierend - elektrische Entladungen zuführen muß, um anschließend auf die Wie-von-selbst-Entstehung der Bausteine des Lebens zu warten, hat er fallengelassen. Aus dem Grund, da sich in hunderten oder tausenden dieser Versuche immer nur Bausteine des Lebens, tote Bausteine ergaben, niemals aber ein Molekül des Lebens gewinnen ließ.

Der Zeitfaktor relativiert den Zufall, was in hundert Jahren ein seltenes Ereignis ist, erscheint in der Dimension von 100 Millionen Jahren als durchaus alltäglich. Und nach der 'Inorganic mineral'-Theorie von Graham Cairns-Smith, auf den sich Dawkins beruft, kann Leben in seinen ursprünglichsten Strukturen in großen Zeitdimensionen aus kristallinen Lehm- oder Tonkörnchen erwachsen sein. Die Kristallstruktur besitzt Speicherfähigkeit; Kristalle sind reproduktionsfähig, und warum sollten nicht immer flexiblere Kristalle Selektionsvorteile erhalten haben? Und ... in einer nach Äonen zählenden Zeitspanne vollzog sich in unendlich kleinen Schritten der Übergang aus anorganischer Materie zu Urformen der RNS oder DNS. Das Leben, aus Staub geformt, geführt in Milliarden Jahren vom Zufall zu all der heutigen Vielfalt; unsere Vernunft, blind und ohne Vernunft entstanden?

Zurück zum Anfang. Bemerkenswert bleibt die Fähigkeit des Menschen, Analogien jeglicher Art zu ziehen, aus Graphiken Biomorphe zu machen und auch von simplen Computerspielereien auf die Entwicklung des Lebens und der Vernunft zu schließen.

Erläuterungen zum Programm UHRMACHER:

Das Programm, geschrieben in ATARI ST-LOGO, entwickelt graphische Gebilde nach neodarwinistischen Prinzipien (s. Dawkins, R., 'The Blind Watchmaker', Penguin 1986).

1. Das "Graphic Display" des LOGO-Interpreters auf maximale Größe einstellen.
2. Programm laden und mit "EVOLUTION" starten.
3. Die 1. Generation bestimmt das Programm selbst über einen Zufallsgenerator, die nächsten Elternformen muß der Anwender auswählen durch Angabe einer Zahl zwischen 1 und 7, entsprechend den 7 Mutationsmöglichkeiten.
4. Links oben wird die Anzahl der Generationen eingeblendet, darunter stehen die Werte für Verzweigung (Z), Strekke (S) und Winkel (W).
5. Gezeichnet werden jedesmal der Ursprung (die 1. Zufallsgeneration), die gewählte Elterngeneration (im Zentrum des Displays) und rundherum ihre 7 Mutationsvarianten.

Für Interessierte:

Auf der letzten Seite seines Buches "The Blind Watchmaker" gibt Richard Dawkins eine Bestelladresse für sein Programm an; leider ist es nur für den Apple Macintosh (oder aladin-Besitzer) zu haben, zum Preis von 31 US$ plus 5,40 US$ Versand- und Verpackungsgebühr:

W W Norton & Company Limited
37 Great Russell Street
London WC1B 3NU

LOGO-Neulinge und langjährige ST-COMPUTER-Leser möchte ich auf den LOGO-Kurs in den Heften 1-3 der ST-COMPUTER 1986 hinweisen, in dem auch der der Algorithmus des Unterprogramms ENTWICKLUNG zu finden ist.

DR. EBELING



ST COMPUTER

Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13

ST COMPUTER

PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

MAXON Computer GmbH
ST-Computer Redaktion
Industriestraße 26
6236 Eschborn

ST COMPUTER

Kleinanzeigen

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13

ST COMPUTER

ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt 13

PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,– bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum Unterschrift

Einzelheft- u. Monatsdisketten Bestellung

ST-Computer können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 6,– (1987) DM 7,– (1988) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli	Aug./Sept.	Okt.	Nov.	Dez.	
											1987 = DM
											1988 = DM
											1989 = DM

+ Gebühr für Porto und Verpackung

(1 Heft DM 2,–, ab 2 Hefte DM 5,–)

☐ Scheck in Höhe zus. DM liegt bei

Disketten Service

Alle Programme, die in ST-Computer veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 ST-Computer-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten (* Monate mit Stern DM 18,–)

* Preis je Diskette 28,– DM	Jan./Febr.	März/Apr.	Mai/Juni	Juli/Aug.*	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.	Mai/Juni
	87	87	87	87	87	87	88	88	88
	Juli* / Aug./Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.	Mai/Juni	Juli/Aug.	Sept./Okt.	Nov./Dez.	
	88 / 88	88	89	89	89	89	89	89	

Lieferung: gegen beigefügten Scheck zuzügl. 5,– DM Versandkosten, unabhängig der bestellten Menge.

Abonnement

Ja, bitte senden Sie mir die ATARI-Computer Fachzeitschrift ab ______
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,– frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,– Normalpost, DM 120,– Luftpost)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name

Vorname

Straße/Nr.

PLZ Ort

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. BLZ

Institut Ort

☐ Ein Verrechnungsscheck über DM ______ liegt bei.

☐ Vorauskasse per Zahlung auf unser Postscheck-Konto Ffm, BLZ 500 100 60, Kto.-Nr. 5537-602

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum Unterschrift Datum Unterschrift



Kleinanzeigen-Auftrag

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an ☐ Hardware ☐ Software **Ich suche** ☐ Hardware ☐ Software ☐ **Tausch** ☐ **Kontakte** ☐ Verschiedenes ☐

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

privat = DM 7,– je Zeile
gewerblich = DM 15,– je Zeile
Chiffregebühr = DM 10,–

Scheck über DM ______ ist beigefügt

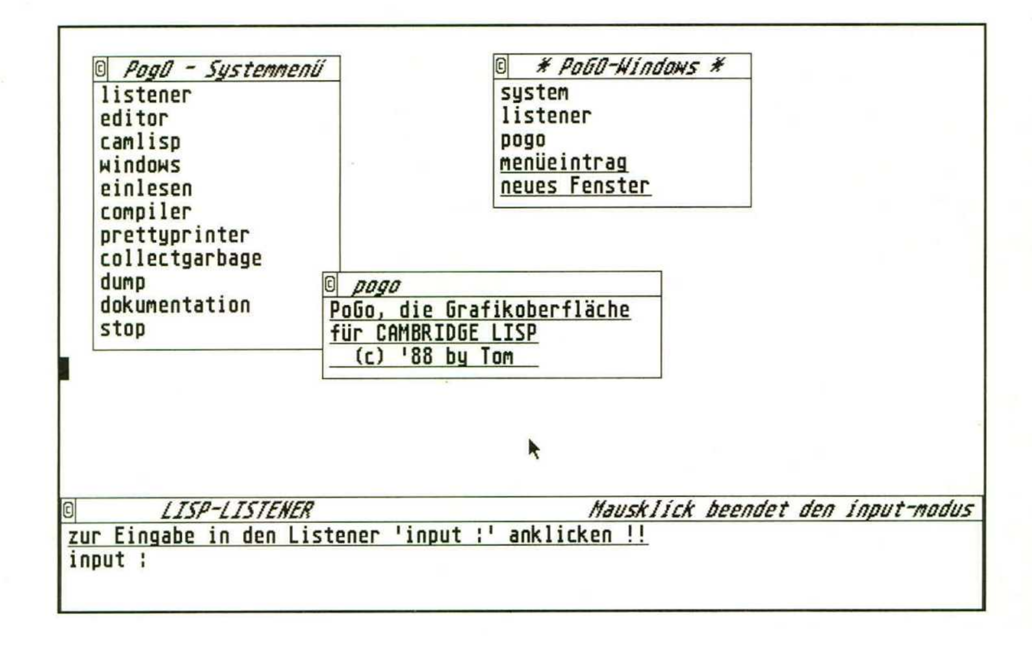
Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum Unterschrift

Absenderangaben auf der Rückseite nicht vergessen

POGO

bringt Lisp zum Tanzen!



Dieser Artikel stellt eine grafische Benutzeroberfläche für die Programmiersprache Lisp vor, die sich vom GEM losgelöst eher an den Oberflächen für Lisp-Workstations orientiert.

ATARI ST-Benutzer können ja über Mangel an Programmiersprachen nicht klagen. Die einzige Sprache, die doch etwas unterversorgt erscheint, ist wohl Lisp.

Zwar gibt es XLISP aus der PD oder das CAMBRIDGE-LISP von MCC, aber so richtig glücklich wird man mit dieser Auswahl kaum.

XLISP schreckt wohl Anfänger wie Fortgeschrittene in Lisp ab, scheinen doch hier alle Vorurteile gegen Lisp bestätigt zu werden, die sich an den Punkten Geschwindigkeit und Übersichtlichkeit festmachen. Und auch das Cambridge-Lisp von MCC, das ja einen durchaus stolzen Preis hat, ist nicht sehr viel komfortabler. Zwar liefert MCC eine MENU-Shell mit, aber Lisp ist eben keine Compilersprache, für die diese Shell eigentlich gedacht ist, sondern gewinnt seinen Reiz gerade auch aus seiner Interaktivität, die man allerdings beim Kontakt mit einem 'nackten' Interpreter noch nicht so recht auskosten kann.

Wenn man mit Lisp mehr anfangen möchte, als z.B. die Fakultät von 10000 zu berechnen, und größere Projekte in Angriff nimmt, benötigt man eine entsprechende Programmierumgebung. Und eine solche möchte ich hier in Form von Pogo für Cambridge-Lisp vorstellen.

Pogo steht für Pseudoparallele objektorientierte Grafikoberfläche. Dahinter verbirgt sich eine grafische Benutzerschnittstelle, die die Kommunikation zwischen Benutzer und Lispsystem auf komfortable und übersichtliche Weise abwickelt, ähnlich wie GEM die Benutzerzugriffe auf das GEMDOS-Betriebssystem verwaltet (s. Bild).

Da GEM-Programmierung in Lisp äußerst unangenehm und auch nicht auf die speziellen Möglichkeiten und Bedürfnisse von Lisp zugeschnitten ist, habe ich eine einfache abstrakte Schnittstelle entworfen, die sich leicht an andere Lispdialekte (und auch an andere Rechner, sofern 'bemaust') anpassen läßt. Alle Funktionen aus Listing 1, die das Wort 'Fenster' enthalten, gehören zu dieser Schnittstelle.

In dieser 'Fenstersprache' (Funktionsdefinitionen stellen in Lisp ja eine Spracherweiterung dar) ist dann die Pogo-Shell geschrieben (s. Listing 1 die Funktionen INIT, START und WORK). Als Applikationen für diese Shell sind dann die restlichen Funktionen zu verstehen (s. Listing 1 sowie insbesondere Listing 2 und 3).

Der Programmierstil, den Pogo benutzt, wird objektorientiert genannt, da die Gewinnung und Verwaltung des Wissens über die jeweilige Verarbeitungsart der einzelnen Fenster nicht vom Hauptprogramm aus geschieht, sondern von den einzelnen abstrakten Objekten, den 'Fenstern', erledigt wird. Das erlaubt hohe Flexibilität für die Applikationen und hält den Code des Hauptprogramms knapp und effizient.

Die Parallelverarbeitung des Pogosystems besteht nun darin, daß beliebig viele 'Fenster' (d.h. 'Fenstersprachen'-programme) gleichzeitig verarbeitet werden können. Das ist natürlich kein echtes Multitasking, aber immerhin ein Pseudomultitasking wie bei den GEM-Deskac-

```
 1:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2:  %                                                   %
 3:  % Pseudoparallelverarbeitende objektorientierte    %
 4:  % Grafikoberfläche PoGo                             %
 5:  % by Thomas Mahler 1988  (c) MAXON Computer GmbH %
 6:  %                                                   %
 7:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 8:
 9:  (setq !*comp t)  %durch Setzen dieser Variablen werden alle
10:                   %Funktiondefinitionen compilert
11:
12:  %%%%%%
13:
14:  (de nth (n list)
15:     (cond ((zerop n) (car list)) (t (nth (sub1 n)(cdr list)))) )
16:
17:  (de nthcdr (n liste)
18:     (cond ((zerop n)liste)
19:           (t (nthcdr (sub1 n)(cdr liste)))))
20:
21:  (de remove (l liste) (delete l liste))%die Funk.nth, nthcdr und remove
22:  %sind in Cambridge Lisp nicht definiert
23:  %%%%%%
24:
25:  (global
26:     '(fensterliste bakliste altliste default path))
27:  (fluid
28:     '(el mx my mcx mcy mbut file  stat clip))
29:
30:     %global und fluid Anweisungen bestimmen die Semantik nicht-lokaler
31:     %Variablen für den Compiler, für das Verständnis nicht wichtig
32:
33:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34:  % Basisroutinen für das GEM-Handling der Fensterfunktionen
35:
36:  (de invers (mode) (cond (mode (v!_rvon 1))(t (v!_rvoff 1))))
37:
38:  (de printat (linie spalte sausdr)
39:     (vs!_curaddress 1 linie spalte)
40:     (princ sausdr))
41:
42:  (de style nil
43:     (printat 1 20 "Welchen Interior-Style? 0,1,2,3:")
44:     (vsf!_interior 1 (read))
45:     (terpri)
46:     (printat 1 20 "Was für ein Fill-Style-Index? 0 - 24 :")
47:     (vsf!_style 1 (read))
48:     (terpri))
49:
50:  (de finit nil      %GEM-Initialisierung!
51:     (vdi!-init)
52:     (aes!-init)
53:     (setq clip (mkvect 3)))
54:
55:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
56:
57:  (de fenster (name)  %zeichnet ein Fenster samt Inhalt,
58:     (vsf!_color 1 1) %führt Vor- und Nachspann aus.
59:     (vswr!_mode 1 1)
60:     (vsf!_interior 1 0)
61:     (cond          %die Fensterliste wird erweitert, falls das
62:        ((not (member name fensterliste)) %Fenster nicht aktiviert war.
63:           (setq fensterliste (append fensterliste (list name)))) )
64:     (cond  %besonders wichtige Fenster werden i.d.
65:        ((and (not (member name bakliste) (fensterhistory name)) %Bakliste
66:           (setq bakliste (append bakliste (list name)))) )%aufgenommen.
67:     (fenstervorspann name) %die Vorspann-Handlung des Fenster ausführen
68:     (setv clip (fensterkoordinaten name) 0)
69:     (v!_hide!_c 1) %vor dem Zeichnen Maus verstecken
70:     (vs!_clip 1 1 clip) %hier wird das Fenster mit
71:     (v!_bar 1 clip)     %Closebox und Titelzeile
72:     (putv clip 3 (iplus (getv clip 1) 17)) %gezeichnet
73:     (v!_bar 1 clip)
74:     (putv clip 2 (iplus (getv clip 0) 10))
75:     (v!_bar 1 clip)
76:     (vst!_color 1 1)
77:     (vst!_effects 1 4)
78:     (v!_gtext
79:        1
80:        (iplus 20 (getv clip 0)) %der Fenstertitel wird in die
81:        (iplus 14 (getv clip 1)) %Titelzeile des Fensters geschrieben
82:        (fenstertitel name))
83:     (vst!_effects 1 0)
84:     (v!_gtext %in die Closebox wird ein (c) gemalt
85:        1
86:        (iplus 2 (getv clip 0))                                  →
```

cessories, allerdings ohne Beschränkung der Anzahl und der Möglichkeit eigene LISPprogramme in dieses System einzubeziehen (z.B. die Fenster *'system*, *'windows* und *'listener*, Listing 1, 2 und 3).

Nun eine kurze Beschreibung der zentralen Funktionen. Die äußere Hülle des Programms, die Funktion INIT, versucht zunächst eine eventuell vorhandene Datei *INIT.LSP* einzulesen, die beliebigen Lispcode enthalten kann (denkbar wäre z.B., sich eine individuelle Pogo-Konfiguration erstellen zu lassen). Dann wird mit OPENWINDOW das Pogo-Systemfenster angemeldet. REFRESH baut anschließend den Bildschirm auf, und die obere Verarbeitungsschleife START wird aufgerufen. Wenn alle Fenster geschlossen sind, übergibt START die Kontrolle wieder an INIT, wo der Benutzer dann wählen kann, ob er das System ganz verlassen möchte, oder ob wenigstens das Systemfenster wieder geöffnet werden soll. Falls aber doch noch geöffnete Fenster vorhanden sind, wird auf ein Mausereignis gewartet; tritt ein solches dann ein, erfolgt die Auswertung des 'Mausklicks':

Wurde ein Doppelklick ausgeführt, wird das Systemfenster neu gezeichnet und zur Verarbeitung gebracht, auch wenn es vorher geschlossen war. Bei einem Einfachklick werden die Fenster in der Reihenfolge ihrer optischen Schichtung auf dem Bildschirm daraufhin durchsucht, ob der Mauscursor sich innerhalb ihrer Koordinaten befindet. Wird dabei kein Fenster gefunden, wird einfach auf den nächsten Klick gewartet. Falls doch ein entsprechendes Fenster gefunden wird, übernimmt die Funktion WORK die eigentliche Verarbeitung des Fensters: Schließen des Fensters bei Anklicken der Closebox, Verschieben auf dem Bildschirm bei Ziehen an der Titelzeile oder Neuzeichen (bzw. Neuberechnen der Fenstergröße) bei Klicken auf die Titelzeile sowie Auswahl des richtigen Fensterelements bei Klicken ins Innere eines Fensters. (Auf weitere Grafikfunktionen wie z.B. Iconoperationen habe ich erst einmal der Einfachheit halber verzichtet).

Wird eine Zeile innerhalb des Fensters angeklickt, übergibt WORK die Kontrolle an die Verarbeitungsmethode des Fensters, die mit der Funktion FENSTERMETHODE gesetzt und ermittelt werden kann. Beim Systemfenster ist dies die Funktion AUFRUF, die die einzelnen Pogo-Applikationen aufruft.

Bei den Fenstern 'windows und 'listener

sind die Verabeitungsmethoden allerdings schon erheblich komplexer (s. Listing 2 und 3). Nachdem die entsprechende Handlung ausgeführt wurde, wartet die Funktion START wieder auf den nächstem Mausklick.

Um zu zeigen, wie die Programmierung der Fenster überhaupt funktioniert, wende ich mich jetzt den 'Fensterfunktionen' zu:

Es existieren insgesamt elf Attribute, die die Art der Verarbeitung eines Fensters bestimmen. Wird als Wert für das Attribut 'nil' angegeben, wird der aktuelle Wert des Attributs ermittelt:

(fensterinhalt 'name'(i1 i2 i3 ...iN))

dem Fenster *name* wird eine Inhaltsliste zugewiesen, wobei Elemente vom Typ STRING bei der Darstellung des Fensters unterstrichen werden.

(fensterflexibel 'name 'on)

Die Größe des Fensters ist flexibel und kann durch einen Klick in der Titelzeile dem jeweiligen Fensterinhalt angepaßt werden.

('off deaktiviert)
(fenstermobil 'name 'on)

Das Fenster läßt sich mit Ziehen der Titelzeile bewegen.

('off deaktiviert)
(fenstertitel 'name "titeltext")

In der Titelzeile erscheint nicht der Name des Fensters, sondern der angegebene Titeltext.

(fensterkoordinaten 'name '
(x1 y1 x2 y2))

Dem Fenster werden Pixelkoordinaten ($x < 640$ $y < 400$) zugeteilt. Falls diese Attributzuweisung schon vor dem ersten direkten Zeichnen des Fensters erfolgt, braucht die Fensterzeichenroutine nicht die Mausposition als oberen linken Punkt zum Ansetzen des Fensters zu benutzen

('resize berechnet die Fenstergröße neu).
(fensterhistory 'name 't)

Ist dieses Attribut gesetzt, wird das Fenster in die Historie des Pogo-Systems aufgenommen und kann im 'windows-Fenster wieder aktiviert werden, falls es vorher geschlossen wurde.

(fensteraktiv 'name)

Aktiviert das Fenster '*name* so, daß es an erster Stelle in der Bearbeitungshierar-

```
 87:         (iplus 14 (getv clip 1))
 88:         (ltoc (ascii 189)))
 89:     (cond         %wenn das Fenster einen definierten
 90:         ((fensterinhalt name)  %Inhalt hat, wird dieser in das Innere
 91:            (schreiben      %des Fensters geschrieben.
 92:               (iplus 5 (getv clip 0))
 93:               (iplus 14 (getv clip 3))
 94:               (fensterinhalt name))))
 95:     (fensternachspann name) %die Nachspann-Handlung wird ausgeführt
 96:     (v!_show!_c 1))           %nach dem Zeichnen Maus sichtbar!
 97:
 98: %%%%%%%
 99: (de maxzeile (liste)     %berechnet die maximale Zeilenlänge, die ein
100:                          %Fenster benötigt
101:     (apply 'max (mapcar liste '(lambda (el) (length (explode el)))) ))
102:
103: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104: %die folgenden Funktionen setzen oder ermitteln
      Eigenschaften der Fenster,
105: %die für deren Verarbeitung wichtig sind.
106:
107: (de fensterflexibel (name flexibel)
108:     (cond
109:         ((eq flexibel 'off)(put name 'flexibel nil)) %die Grösse des F.
                                                          ist starr
110:         (flexibel(put name 'flexibel t))%die Grösse des F. ist flexibel
111:         (t (get name 'flexibel)))) %ermittle den Flexibelwert des F.
112:
113: (de fenstermobil (name mobil)
114:     (cond
115:         ((eq mobil 'off) (put name 'mobil nil))%Fenster lässt sich
                                                    nicht bewegen
116:         (mobil (put name 'mobil t)) %Fenster lässt sich verschieben
117:         (t (get name 'mobil)))) %ermittle den Mobilwert des Fensters
118:
119: (de fenstertitel (name titel) %setze oder ermittle den Titelschriftzug
120:     (cond     %des Fensters
121:         (titel (put name 'titel (ltoc titel)))
122:         (t (cond ((get name 'titel))(t (ltoc name)))) ))
123:
124: (de cls nil (v!_clrwk 1))          %clearscreen
125:
126: (de fensterkoordinaten (name koords)
127:     (cond
128:         ((not koords) %ermittle die Koordinaten eines Fensters:
129:            (cond
130:               ((get name 'koords)) %die Koordinate sind bekannt
131:               (t (vq!_mouse 1 mbut mcx mcy)%die Koordinaten sind nicht
                                                  bekannt
132:                  (putv clip 0 mcx) %das Fenster wird dorthin gerechnet,
133:                  (putv clip 1 mcy) %wo die Maus sich befindet, mit der
134:                  (putv        %für den Fensterinhalt
135:                     clip        passenden %Grösse.
136:                     3
137:                     (iplus
138:                        (getv clip 1)
139:                        (iplus
140:                           25
141:                           (itimes 16 (length(fensterinhalt name)))) ))
142:                  (putv
143:                     clip
144:                     2
145:                     (iplus
146:                        (getv clip 0)
147:                        (iplus
148:                           10
149:                           (itimes
150:                              8
151:                              (maxzeile
152:                                 (cons
153:                                    (fenstertitelname)
154:                                    (fensterinhaltname)))) )))
155:                  (fensterkoordinaten %die Koordinaten wurden errechnet
156:                     name  %und werden jetzt gespeichert
157:                     (list
158:                        (getv clip 0)
159:                        (getv clip 1)
160:                        (getv clip 2)
161:                        (getv clip 3)))) ))
162:         (t (cond  %falls das Schlüsselwort 'resize angegeben wurde,
163:               ((eq koords 'resize) %wird die Grösse des Fenster neu
                                        berechnet
164:                  (fensterkoordinaten
165:                     name
166:                     (list
167:                        (putv clip 0 (nth 0(fensterkoordinaten name)))
```

```
168:                 (putv clip 1 (nth 1 (fensterkoordinaten name)))
169:                 (iplus
170:                    (getv clip 0)
171:                    (iplus
172:                       10
173:                       (itimes
174:                          8
175:                          (maxzeile
176:                             (cons
177:                                (fenstertitelname)
178:                                (fensterinhaltname)))) ))
179:                 (iplus
180:                    (getv clip 1)
181:                    (iplus
182:                       25
183:                       (itimes
184:                          16
185:                          (length(fensterinhalt name)))) ))) )
186:           (t (put name 'koords koords)))) ))%setze d. Koordinaten
                                                    des Fensters
187:
188:  (de fensterinhalt (name inhalt)
189:     (cond (inhalt (put name 'inhalt inhalt)) %setze d. Inhalt eines
                                                  Fensters
190:           (t (get name 'inhalt))))%ermittle den Inhalt eines Fensters
191:
192:  (de fensteraktiv (name)
193:     (cond
194:        (name     %aktiviere das angegebene Fenster
195:           (setq fensterliste (cons name (remove name fensterliste)))
196:           (setq altliste (cons name (remove name altliste))))
197:        (t (cond %ermittle das aktivierte Fenster
198:              ((consp fensterliste) (car fensterliste))
199:              (t 'ende)))) )
200:
201:  (de fenstermethode (name methode)
202:     (cond
203:        (methode (put name 'methode methode)) %setze d. Verarbeitungs-
                                                  methode
204:        %des Fensters
205:        (t (get name 'methode)))) %ermittle die Methode des Fensters
206:
207:  (de verarbeitefenster (name but) %führe die Methode des Fensters aus
208:     (cond  %falls keine Methode bekannt ist
209:        ((fenstermethode name) (errorset (fenstermethode name) 0))
210:        (t (setq but  %eine Warnung ausgeben
211:              (form!_alert
212:                 1
213:                 (ltoc
214:     "[1][Die Verarbeitungsmethode|für dieses Fenster ist
           |nicht bekannt !!!] [peinlich...]")))) ))
215:
216:  %%%%%%%%
217:  (global '(linie))
218:
219:  (de cut (exp zw) %cut schneidet alles ab, was die
220:     (setq zw (explode (ltoc exp))) %Funktion V!_GTEXT zum Abstürzen
                                          bringen würde
221:     (cond
222:        ((lessp 79 (length zw))
223:           (rplacd (nthcdr 78 zw) '(!"))
224:           (ltoc (compress zw)))
225:        (t (ltoc exp))))
226:
227:  (de schreiben (x y text)
228:     (cond  %diese Funktion erledigt das Schreiben der Fensterinhalte
229:        ((consp text)
230:           (cond ((stringp (car text))
              (vst!_effects 1 8)))
231:           (v!_gtext 1 x y (cut (car text)))
232:           (vst!_effects 1 0)
233:           (schreiben x (setq linie (iplus y 16))(cdr text)))))
234:
235:
236:  (de schliessefenster (name) %das Fenster wird von der Fensterliste
                                  entfernt
237:     (setq fensterliste (remove name fensterliste)) %und der Abspann
                                                       ausgeführt
238:     (abspann name))
239:
240:  (de fensterhistory (name history)
241:     (cond
242:        (history (put name 'history history)) %dieses F. wird in die
                                                  Bakliste
243:        %aufgenommen
244:        (t (get name 'history)))) %ermittelt den Historywert      →
```

chie der Fenster steht. Diese Funktion bestimmt die Reihenfolge der Fensterverarbeitung bei der Suche nach dem Fenster, in dem der Mausklick stattfand.

(fenstermethode 'name 'Sexpr)

Dieses ist die wichtigste Funktion für die Multitaskingprogrammierung, denn sie zeigt dem Pogo an, wie das Fenster nach der Auswahl einer bestimmten Fensterzeile weiterverarbeitet werden soll. Aber es ist einfacher als es vielleicht klingt:

Pogo übernimmt alle Aspekte der Fensterverwaltung bis zu der Stelle, an der ein Element des Fensterinhalts per Anklicken ausgewählt wurde. Dieses Element wird in der fluid-Variablen *EL* abgelegt, auf die dann die Pogo-Applikationen zugreifen.

Zum Beispiel so:

(fenstermethode 'pogo '(eval EL)).

Das ist genau die Methode nach der das Systemfenster seine Programme ausführt.

(fenstervorspann 'name 'Sexpr)

Es kann manchmal nützlich sein, Handlungen auszuführen, die sich auf ein Fenster beziehen,o hne daß das Fenster selbst aktiviert ist, z.B. wenn ein veränderlicher Fensterinhalt ständig aktualisiert werden und die Fenstergröße dem Inhalt angepaßt werden soll.

Bei jedem REFRESH wird für alle Fenster der jeweilige Vorspann 'sexpr ausgeführt.

(fensternachspann 'name 'sexpr)

Wenn ein Vorspann nützlich ist, kann es evtl. ja auch ein Nachspann sein. Der Nachspann *'sexpr* wird bei jedem REFRESH nach Zeichnen jedes einzelnen Fensters ausgeführt.

(abspann 'name 'sexpr)

Wenn das Fenster durch Anklicken der Closebox geschlossen wird, wird die Nachspannhandlung *'sexpr* ausgeführt.

Nach dieser eher technischen Beschreibung gehe ich abschließend noch darauf ein, was Pogo für den Benutzer im einzelnen bietet. Hat man alle Quelltexte in die Dateien POGO, WINDOWSX und LISTENER (Listing 1, 2 und 3) geschrieben und auf der Lispdiskette neben LISP.PRG abgespeichert, startet man das Cambridge-Lisp.

```
245:
246:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
247:  %die drei folgenden Funktionen bestimmen Handlungen, die vor und nach
dem
248:  %Zeichnen sowie nach dem Schliessen eines Fensters ausgeführt werden
249:
250:  (de fenstervorspann (name vorspann)
251:     (cond
252:        (vorspann (put name 'vorspann vorspann)) %setze Vorspann
253:        (t (errorset (get name 'vorspann) 5)))) %führe Vorspann aus
254:
255:  (de fensternachspann (name nachspann)
256:     (cond
257:        (nachspann (put name 'nachspann nachspann)) %setze Nachspann
258:        (t (errorset (get name 'nachspann) 5)))) %führe Nachspann aus
259:
260:  (de abspann (name abspann)
261:     (cond
262:        (abspann (put name 'abspann abspann)) %setze Abspann
263:        (t (errorset (get name 'abspann) 5)))) %führe Abspann aus
264:
265:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
266:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
267:  %die folgenden Funktionen bilden den Kern des PoGo-Systems
268:  %START wartet auf Mausereignisse, berechnet die Hierarchie der Fenster
269:  %neu und übergibt WORK den Namen des aktivierten Fensters und dessen
   Koordinaten
270:
271:
272:  (de start nil
273:     (prog (mx my mcx mcy mbut mstat)
274:       loop %Beginn der obersten Verarbeitungsschleife
275:        (cond
276:           ((null fensterliste) (return)) %falls alle Fenster
                                            geschlossen sind
277:              %beende START
278:           (t (evnt!_button 2 1 1 mcx mcy mbut stat) %warten auf
                                                      Mausereignis
279:              (cond
280:                 ((eq mbut 3) (fenster 'system)) %bei Doppelklick
                                                   aktiviere das
281:              %Fenster 'System
282:                 (t (prog (liste)
283:                       (setq liste fensterliste)
284:                 schleife %durchforste die Fensterliste
285:                       (cond %nach dem 'Angeklickten'
286:                          ((or
287:                               (null liste)
288:                               (cond
289:                                  ((innerhalbp %falls die Maus ein F.
290:                                  mcx %angeklickt hat, übernimmt
291:                                  mcy %work die Weiterverarbeitung
292:                                  (fensterkoordinaten (car liste)))
293:                                     (fensteraktiv (car liste))
294:                                     (work
295:                                        (fenster koordinaten
296:                                           (fensteraktiv))
297:                                        (fensteraktiv))
298:                                     t)))
299:                            (return 'done)))
300:                       (setq liste (cdr liste))
301:                       (go schleife)))) ))
302:        (go loop)))
303:
304:  (de refresh nil %baut den Bildschirm neu auf, die Reihenfolge der F.
305:     (cls)      %wird umgekehrt, damit sich d. optische Schichtung mit
306:                %der Verarbeitungslogik deckt
307:     (mapcar (reverse fensterliste) '(lambda (f)(fenster f))))
308:
309:
310:  (de work (koords aktiv breite höhe) %work greift auf die Variablen
                                         mcx u. mcy zu
311:     (cond
312:        ((lessp mcy (iplus 17 (nth 1 koords))) %ging der Klick in die
                                                 Titelzeile?
313:           (cond
314:              ((lessp mcx (iplus 10 (nth 0 koords))) %falls Mausklick
                                                      in Close-
315:                 (schliessefenster aktiv) %box, Fenster schliessen
316:                 (refresh))
317:              (t (vq!_mouse 1 mbut mx my)
318:                 (cond %wurde das F.nur kurz angeklickt wird seine
319:                       %Grösse neu berechnet, falls es 'flexibel' ist
320:                    ((and (zerop mbut) (fensterflexibel aktiv))
321:                       (fensterkoordinaten aktiv 'resize)
322:                       (cond                                         →
```

Sobald der Input-Prompt erscheint, gibt man (rdf 'POGO)(init)<return> ein. Das gesamte Programm wird compiliert, und das Pogo-Systemfenster erscheint. Dieses Fenster stellt dem Benutzer jetzt einige wichtige Funktionen zur Verfügung, die durch Anklicken ausgewählt werden:

Editor: Diskettendateien können ediert und sofort evaluiert werden.

Listener: Der ins Pogo integrierte Lisp-Listener. Wird eine der Inhaltszeilen angeklickt, geht der Listener in den Input-Modus, in dem er sich wie ein Interpreter bedienen läßt, mit dem Unterschied, daß ein einfacher Mausklick ausreicht, den Input-Modus wieder zu verlassen.

Windows: die Pogo-Applikation, welche das Erzeugen neuer Fenster, das Verändern der Flexibel-, Mobil- und Inhalt-Attribute sowie das Wiederaufrufen alter abgebrochener Prozesse erlaubt.

Camlisp: Für manche Anwendungen wird man den altbewährten Cambridge-Lisp-Interpreter bevorzugen, der mit dieser Funktion aufgerufen wird.

Einlesen: Diskettendateien können zur Evaluation ausgewählt werden.

Compiler: Die Art der Evaluation, Compilierung oder Interpretierung neuer Funktionsdefinitionen wird bestimmt.

Dokumentation: Die Online-Dokumentation kann, falls vorhanden, studiert werden; Stichwortsuche wird unterstützt.

Collectgarbage: Komfortable Bedienung des GarbageCollectors von Pogo aus.

Dump: Es kann eine Bootdatei LISP-ROOT erzeugt werden, die den aktuellen Lispzustand sichert und beim Neustart von Lisp wieder die Pogo-Oberfläche installiert.(Es sollten ca. 130k Platz auf der Diskette sein.)

Stop: Cambridge-Lisp wird verlassen, Rückkehr zum GEM-Desktop.

Hat man Pogo einmal installiert, kann man es einfach verbessern und erweitern, indem man dem Systemfenster neue Menüpunkte zuweist und eigene Fensterprogramme schreibt, für die der Listener oder das Windowsfenster einige der wichtigsten Vorgehensweisen demonstrieren. Wenn man häufiger mit Pogo arbeitet, ist es zweckmäßig, sich mit der Option DUMP eine Bootdatei erstellen zu lassen, die Pogo nach dem Lispstart aufruft, ohne erst mühselig alles compilieren zu lassen.

Thomas Mahler

```
323:                    ((innerhalbp
324:                         (nth 2 koords)
325:                         (nth 3 koords)
326:                         (fensterkoordinaten
                              aktiv))
327:                         (fenster aktiv))
328:                     (t (refresh))))
329:              (t (cond %falls Maustaste
                   gedrückt bleibt, und das F.
330:                     ((fenstermobil aktiv)
                  %'mobil' ist, wird es auf
331:                         (graf!_dragbox
                  %dem Bildschirm verschoben
332:                             (setq breite
333:                                 (idifference
334:                                     (nth 2
                                          koords)
335:                                     (nth 0
                                          koords)
                                         ))
336:                             (setq höhe
337:                                 (idifference
338:                                     (nth 3
                                          koords)
339:                                     (nth 1
                                          koords)))
340:                             (nth 0 koords)
                                 (nth 1 koords)
                                  0 18 2000
341:                              2000 mx my)
342:                         (fensterkoordinaten
343:                             aktiv
344:                             (list
345:                                 mx
346:                                 my
347:                                 (iplus mx
                                      breite)
348:                                 (iplus my
                                      höhe)))
349:                         (refresh))
350:                     (t (fenster aktiv)))) ))))
351:         ((lessp %falls ins Innere des Fensters
                    geklickt wurde, ermittle
352:              mcy %d. genauen Menüpunkt u.verabeite
                    das Fenster nach der
353:           (iplus  %angegebenen Methode
354:              (nth 1 koords)
355:              16
356:              (itimes 16 (length (fensterinhalt
                   aktiv)))) )
357:          (setq el
358:             (nth
359:                (iquotient
360:                    (idifference mcy (iplus 16
                        (nth 1 koords)))
361:                    16)
362:                (fensterinhalt aktiv)))
363:          (verarbeitefenster aktiv))))
364:
365: (de zwischenp (a x y) %liegt ein Wert zwischen
                          zwei anderen?
366:    (cond
367:       ((or
368:            (and (lessp x a) (lessp a y))
369:            (and (lessp a x) (lessp y a)))
370:          t)
371:       (t nil)))
372:
373: (de innerhalbp (x y koli) %liegt ein Punkt
                          innerhalb eines Rechtecks?
374:    (cond
375:       ((and
376:            (zwischenp x (nth 0 koli) (nth 2 koli))
377:            (zwischenp y (nth 1 koli) (nth 3 koli)))
378:          t)
379:       (t nil)))
380:
381: (de aufruf (el) %Aufruf ist die
                 Verarbeitungsmethode des F.
382:          %'system falls eine Funktion nicht
               bekannt ist
383:          %wird versucht eine entsprechende
               Datei von der Diskette
384:          %einzulesen, ansonsten wird die
               Funktion ausgeführt.
385:    (cond                                   →
```

```
386:       ((not (boundp el))
387:          (errorset '(rdf (compress
             (cons el '(xyz)))) 0)
388:          (refresh)))
389:    (errorset (list el) 0))
390: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
391: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
392:
393: %Jetzt folgen Funktionen die vom `SYSTEM Fenster
      aus aufgerufen werden können
394:
395: (de compiler (but)  %die Evaluationsmethode,
                    Compilierung oder Interpretation
396:    (setq but   %neuer Funktionsdefinitionen kann
                    festgelegt werden
397:       (form!_alert
398:          1
399:          (ltoc
400:    "[2][sollen alle weiteren|Funktionsdefini-
     tionen|compiliert werden ?][compiler|
     interpreter]")))
401:    (cond
402:       ((eq but 1) (setq !*comp t))
403:       ((eq but 2) (setq !*comp nil))))
404:
405:
406: (de collectgarbage () %diese GC-Routine lässt sich
                          vom Systemfenster
407:    (graf!_mouse 2 0)  %aus aufrufen
408:    (verbos 0)
409:    (reclaim)
410:    (graf!_mouse 0 0))
411:
412: (de Camlisp () %übergibt die Kontrolle an den
                    gewohnten Camlisp-Interpreter
413:    (cls)          %zurück zu PoGo mit FIN
414:    (printat 1 1 "Cambridge Lisp Supervisor ,
                     zurück zu PoGo mit Fin!")
415:    (terpri)
416:    (cursoron)
417:    (supervisor)
418:    (cursoroff)
419:    (refresh))
420:
421:
422: (de dokumentation nil %falls eine Dokumentation
                          auf der Diskette ist,
423:    (cursoroff)       %lässt sich diese studieren
424:    (v!_hide!_c 1)
425:    (edit 'dokument)
426:    (refresh))
427:
428: (fluid '(but file))
429:
430: (de editor (but ) %eine Diskettendatei wird
                      ausgewählt, ediert und kann
431:    (cond         %anschliessend evaluiert werden.
432:       ((objektauswahl default)
          %die Fileselectboxverwaltung
433:          (cursoron)
434:          (v!_hide!_c 1)
435:          (edit file)
436:          (cursoroff)
437:          (v!_show!_c 1 0)
438:          (evalask file)))
439:    (refresh))
440:
441: (de dump (but) %es kann ein LISPDUMP erzeugt
                   werden,
442:    (setq but  %lisp startet dann wieder mit Pogo
443:       (form!_alert
444:          2
445:          (ltoc
446:             "[2][Soll tatsächlich ein neues|
              LISPROOT-File erzeugt werden?]
              [nur zu|nee lieber nicht]")))
447:    (cond ((eq but 1)(preserve 'init))))
448:
449:
450: (de einlesen (but)  %Diskettendateien können
                        evaluiert werden
451:    (cond ((objektauswahl default) (rdf file)))
452:    (refresh))
453:
454: (global '(n s))
455: (fluid '(but))                                 →
```



STARKE SOFTWARE

ST-KREATIV-DESIGNER
DAS SUPER-PROGRAMM ZUM KREATIVEN ERSTELLEN VON GRAFIKEN ALLER ART – OHNE MALKENNTNISSE

Das Selbstverständliche
Einfache Bedienung. Ansprechende Menüs. Alle Bildoperationen lassen sich rückgängig machen. Automatischer Malmodus, der Muster in Millionen Variationen erstellt.
- 2 Füllroutinen
- 2 Blockspeicher
- Absolut flimmerfreie Block- und Bildverarbeitung. Die eingestellte Verknüpfung ist schon beim Zeichnen und bewegen von Bildteilen zu sehen
- WYSIWYG! Blocks lassen sich in jeder Richtung über den Bildrand hinausschieben. Ausschneiden und Einkopieren von beliebigen Formen!
- Freihand: verschiedene Pinsel. Füllmuster als Pinsel, Malen mit Bildausschnitten. Spiegeln beim Zeichnen um 1 oder 2 Achsen
- Füllmuster definieren ohne Editor – eine ganz neue einfache aber wirkungsvolle Methode
- Vergrößern (Bild oder Text)
- Verkleinern, 3 Endprodukte zur Auswahl
- Spiegeln
- Drehen: dreht in 2 Richtungen gleichzeitig
- Stauchen
- Rahmen(1): Erzeugen von Rahmen aus allen Graphiken
- Rahmen(2): automatische Rahmenberechnung
- Schatten: automatische Schattenberechnung (3D Effekt)
- Teilmuster: aus Bildteilen (und Teilchen) können weitere Muster, auch Füllmuster erstellt werden
- Font's: 23 Größen, 21 Arten, 4 Verknüpfungen, weitere Verarbeitung durch Schatten. Rahmen usw. möglich. Schreibrichtung und Farbe kann während des Schreibens geändert werden
- Lineal: Einblendbare Einteilung
- Radierer und Sprühdose in jeder Größe einstellbar. Zwei Sprühdosen Modi
- Schnelle Lupe mit: Punkt, Rechteck, Linie, Inventieren, Löschen
- Weitere Optionen: z. B. Bewegen, Kopieren, Kopieren nach..., Rechteck, Kreis, Linie, 3D-Rechteck, Preistafel...
- Invertieren auf Knopfdruck während gezeichnet wird
- Voll Mausgesteuert
- Kreativ-Designer – Bilder können in Signum 2 verarbeitet werden!
- Für Designer, Werbestudios, Druckgewerbe, Graphiker, aber auch für den nicht professionellen Anwender z. B. zum Erstellen einer Schülerzeitung, Handzettel, Grußkarten usw.

Das Besondere
- Fragen Sie mal die Bedienungsanleitung Ihres Zeichenprogramms ob folgende Funktionen möglich sind:
- Selbständige Hintergrund- und Füllmustererzeugung
- Über 200 Selbstdefinierbare Füllmuster im Speicher
- Kein Flimmern kein Gummiband
- Beim Zeichnen und Bewegen ist immer das fertige Endprodukt zu sehen
- Einige selbständige Rahmen und Schattenberechnungen
- Drei Vier Fünf... Neunundvierzigecke
- Automatisch Blocks ausschneiden
- Block paßgenau Einkopieren mit oder ohne Rahmen
- Blocks lassen sich in jede Richtung über den Bildrand hinausschieben
- Blaupause, Abmalen, Durchmalen von anderen Bildern
- Ein echtes Zeichenbrett mit Linealen
- Bilder maskieren
- Blocksatz, Proportionalsatz, Zentrieren

DM 128,–

SALIX DIALOGEXPERTE
Wer schon einmal eine Anwendung unter GEM geschrieben hat – mit Drop-down-Menüs, Dialogboxen und Alerts, wird es wissen:
So einfach sich die einzelnen Objekte mit dem Resource Construction Set Editor erzeugen lassen, so umständlich ist ihre Einbettung in das Anwendungsprogramm.
Es geht auch anders: SALIX hat jetzt einen DIALOG-Experten entwickelt, der die RSC- und die DEF-Dateien selbständig untersucht. Dann geht's erst richtig los: **„Programming by doing"** heißt die Device. Zunächst werden die Fenster der Anwendung definiert – mit der Muas. Dann wird die Bedeutung der Menüpunkte festgelegt, einfach indem man sie anklickt und das aufzurufende Prädikat eingibt.
Mehrere Menüleisten sind zulässig. Die einzelnen Menüpunkte lassen sich in logische Gruppen zusammenfassen, das Check-Symbol wird kontrolliert, und und und...
Die Knöpfe der Dialogboxen und Alerts lassen sich symbolisch ansprechen, eine Reihe von Prädikaten ermöglicht den einfachen Zugriff auf die GEM-Objekte von der Anwendung aus, komplexe Formulare können auch dynamisch aus der Anwendung heraus aufgebaut werden.
Sie haben die Wahl: sich weiterhin damit abzuquälen, Menüpunkte auszuzählen, oder sich lieber auf das Design und Logik Ihrer Anwendung zu konzentrieren.
Der Dialogexperte läuft unter SALIX PROLOG 2.6.

DM 79,–

ST-MAXIDAT
DIE DATEIVERWALTUNG FÜR DEN HOHEN ANSPRUCH
- Schnittstelle zu Textverarbeitungsprogrammen für Serienbriefe, Rechnungen, Mahnungen, ... (möglich mit allen ASCII-Editoren (1st Word, Tempus, ...); große Möglichkeiten, da z. B. Datumsausgabe und Zugriff auf externe Textdatei)
- Schnittstelle zum Zeichenprogramm 'Kreativ-Designer' für Bilddatenverwaltung (Diashow für Werbezwecke, IC-Datenbank mit abgebildetem IC, Adressveraltung mit Foto/Stadtplan, bebildeter Anleitung) Formate: STAD/Neochrom/Degas
- frei erstellbare Druckerformate für beliebige Formulare, Etiketten, Listen, ...
- Rechnen im Datensatz (+ – ★/, Prozent) z. B. zur automatischen Ermittlung der MwSt sowie Bilanzvergleich (Gesamtergebnis, Durchschnitt)
- Druckeranpassung für alle Drucker im laufenden Programm erstellbar (Init, Exit, Drucktabelle für Umlaute, Druckersteuerung und sonstiges)
- mehrere Dateien verknüpfbar und daraus durch Selektierung neue erstellbar
- 'Filter' zur Beschränkung der Ausgabe von Datensätzen mit bestimmten Kriterien
- Automatisches Löschen von bestimmten Datensätzen (z. B. alle Kunden, die seit 1986 keine Bestellung mehr aufgegeben haben)
- Übersichtsdarstellung mit verschiedenen Zeichengrößen
- Suchen und Sortieren nach allen Datenfeldern sowie globales Suchen
- leistungsfähiger Editor zur Beschriftung der Datensätze
- Prg. arbeitet arbeitsspeicherorientiert, daher hohe Verarbeitungsgeschwindigkeit
- Programmaufruf aus dem laufenden Programm ohne Datenverlust
- konvertiertes Ausdrucken eines beliebigen Textes mit Zeilennr., Dateiname, Datum
- Uhrzeit und Datum neu stellbar
- Feldnamen jederzeit änderbar
- max. 100 000 Datensätze je Datei bei Mega ST-4
- voll GEM-gesteuert und einfachste Bedienung trotz vieler Funktionen

DM 98,–

SALIX-PROLOG
leistungsfähiges KI-Programmiersystem
- Edinburgh-Standard
- schnell, ca. 1200 LIpS
- ca. 160 eingebaute Funktionen
- Gleitkommaarithmetik
- läuft unter GEM
- ca. 140 GEM Library Funktionen
- Datenbankeditor
- Einbindung von Fremdeditoren
- zyklische Strukturen werden verarbeitet
- Exception Handling
- leistungsfähiges Testsystem

Jetzt neu: Version 2.6
- benutzerdefinierte Funktionen
- globale Variable
- Clipboard-Device

DM 198,–

Update von 2.0 – 2.5 DM 49,–

ST DISK BOX
DISKETTEN-VERWALTUNGSPROGRAMM
- Ablegen der Disketten nach Nummern (650 Stück)
- Ablegen in freibenennbaren Sparten
- Mitablegen von Texten möglich
- Suchen nach Programmname, Disknummer, Text, Extender
- Löschen durch Mausklick
- Druckfunktion für Disknummer, Extender, Sparte
- Programm erkennt doppelte Einsortierung in eine Datei
- Programm ist voll GEM-unterstützt
- große Geschwindigkeit
- komfortabel zu bedienen
- Nachträgliches Bearbeiten der Dateien möglich

DM 49,–

ST-C.A.R.
Computer Aided Regulation
Das Programm zur Systemanalyse und Regelungssynthese

Zielgruppe / Aufgabenbereich
Studenten der Ingenieurwissenschaftlichen Fachrichtung
- Veranschaulichung der Auswirkung von Regelkreisglieder/-strecken
- Überprüfung erarbeiteter Lösungen (numerisch und grafisch)
- Einsatz in der Entwicklung eigener Regelsysteme (z. B. Diplomarbeit)

Ingenieure der technischen Fachrichtung
- Einsatz in der Entwicklung benötigter Regelkreise (Zeitersparnis)
- Systemanalyse vorhandener Regelungen (Untersuchungen möglicher Änderungen)
- Unterstützung beim Entwurf durch 'weniger Erfahrene' (Auffrischung des Wissens, eigenes Wissen überprüfen)

Institute, Lehr- und Forschungseinrichtungen des technisch-wissenschaftlichen Bereiches
- Einsatz in der Entwicklung benötigter Regelsysteme
- Analyse vorhandener Regelsysteme (Untersuchung möglicher Änderungen)
- Hilfe bei der Erstellung von Lehr-(Unterrichts-)Material
- Direkter Einsatz im Unterricht ('am lebenden Objekt')

C.A.R. ...das Programm mit umfassender Leistung
C.A.R. ...ausführliche Anleitung mit Beispielen
C.A.R. ...das Programm für professionelle Ansprüche

DM 198,–

ST PRINT
DAS VIELFACH BEWÄHRTE MULTIACCESSORY JETZT IN ERWEITERTER VERSION V2.1 INCL. VIRUSKILLER

ST-PRINT erleichtert durch seine Vielseitigkeit und einfache Bedienung die tägliche Arbeit. Insbesondere Anpassungs- und Verständigungsprobleme zwischen Computer und Drucker werden gelöst.
Parameter -load und -save alle Einstellungen (Ramdisk, Spooler, Hardcopy und Druckervoreinstellung) werden abgespeichert und können auch beim Booten automatisch eingestellt werden.

1. Resetfeste Ramdisk
- Größe einstellbar von 32 – 4000 KB, auf Wunsch resetfest oder abschaltbar
- kann auf Laufwerk C bis P gelegt werden
- arbeitet problemlos mit einer Harddisk zusammen

2. Druckerspooler
- Größe einstellbare von 2 – 510 KB
- abschaltbar
- arbeitet mit TOS- und GEM-Programmen
- auch für Hardcopies
- in Maschinensprache programmiert und interruptgesteuert, dadurch immer optimale Geschwindigkeit beim Ausdruck
- Größe nun 2 – 4000 kByte

3. Druckervoreinstellung
- komfortable Druckereinstellung mit der Maus: Knopfdruck statt Handbuch oder DIP-Schalter
- viele Einstellmöglichkeiten: Zeilenvorschub, Schriftart, Zeichensatz, Papierränder etc.
- Einstellung des Druckers vom Desktop und aus jedem GEM-Programm (VIP-Professional, Wordplus, Tempus etc.) möglich
- **kann an alle Matrixdrucker angepaßt werden, diverse Druckertreiber gehören zum Lieferumfang, auch eigene Anpassungen können erstellt werden**
- **Druckeranpassung jederzeit nachladbar**

4. Druckerzeichenkonverter
- Anpassung aller Atari-Zeichen an den verwendeten Drucker, ermöglicht den korrekten Ausdruck von Umlauten und Sonderzeichen auf allen Matrixdruckern
- Umlaute und Sonderzeichen können gemischt verwendet werden
- endlich problemloser Ausdruck mit Programmen (TOS oder GEM), die keine eigene Druckeranpassung haben

5. Hardcopyroutine
- nutzt die Fähigkeiten von 9-, 18- und 24-Nadeldruckern
- kann an jeden Matrixdrucker angepaßt werden
- verschiedene Auflösungen, Schnelldruck bis Qualitätsdruck
- arbeitet mit dem eingebauten Spooler zusammen *gespoolte Hardcopy
- Auslösung der Hardcopy durch Alternate/Help, kann genauso wieder abgebrochen werden
- Umsetzung der Farben in Grauwerte (im MID-RES Modus)
- nun auch Ansteuerung HP-Laserjet-kompatibler Laserdrucker
- Hardcopy als Bild auf Diskette, Ramdisk oder Harddisk im Degas- oder Screenformat

6. Viruskiller
- Erkennen und Vernichten von BOOT-SEKTOR-Viren
- GEM-Programm, durch einfaches Umbenennen (.ACC statt .PRG) auch als Accessory verwendbar

DM 69,–

DM 19,–

* alle Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON
an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ______________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

```
456:
457:  (de objektauswahl (name)%diese Funktion übernimmt
                              die Verwaltung der
458:     (setq n (mkbuff 64)) %GEM-Fileselectbox
459:     (setq s (mkbuff 64))
460:     (setbuff n (ltoc name))
461:     (setbuff s (ltoc path))
462:     (fsel!_input s n but)
463:     (cond
464:        ((eq but 1)
465:           (setq s (ctol s))
466:           (clearbuff)
467:           (pack s)
468:           (setq path (ltoc (mkatom)))
469:           (setq n (ctol n))
470:           (clearbuff)
471:           (pack n)
472:           (setq file (mkatom))
473:           (setq default (ltoc file))
474:           file)
475:        (t nil)))
476:
477:  (unglobal '(n s))
478:
479:  (de evalask (file) %diese Funktion fragt den
                         Benutzer nach der gewünschten
480:     (setq but        %Evaluationsmethode
481:        (form!_alert
482:           1
483:           (ltoc
484:      "[2][soll das File |evaluiert werden?][inter
                            prete|compile|nein]")))
485:     (cond
486:        ((eq but 1) (setq !*comp nil) (rdf file))
487:        ((eq but 2) (setq !*comp t) (rdf file))))
488:
489:  (de openwindow nil  %diese Funktion meldet das
                           'SYSTEM-Fenster an
490:     (fenstertitel 'system "PogO - Systemmenü")
491:     (fenstermobil 'system 'on)
492:     (fensterflexibel 'system 'on)
493:     (fenstermethode 'system '(aufruf el))
494:     (fensterhistory 'system t)
495:     (fensteraktiv 'system))
496:
497:  (de init nil   %die äussere Hülle des Programms,
                       die Funktion (INIT)
498:     (errorset '(rdf 'init!.lsp) 5)
         %wird zum Starten von PoGo aufgerufen
499:     (openwindow) (graf!_mouse 0 0) (refresh)
         (start)
500:     (setq but
501:        (form!_alert
502:           1
503:           (ltoc
504:              "[2][Willst Du POGO|wirklich
              verlassen?][ ja |nein]")))
505:     (cond ((eq but 2) (init)) (t (stop))))
506:
507:  (fensterinhalt 'system '(editor listener windows
                             camlisp einlesen compiler
508:                            dokumentation
                            collectgarbage dump stop))
509:  %der Fensterinhalt des Fensters SYSTEM wird ange.
510:
511:  (finit)
512:  (terpri)
513:  (printc "bitte (INIT) eingeben um PoGo zu
             starten!")
514:  (terpri)
515:
516:  fin
```

```
 1:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2:  %                           %
 3:  %          WINDOWSX         %
 4:  %            PoGo           %
 5:  %      by Thomas Mahler     %
 6:  % (c) MAXON Computer GmbH %
 7:  %                           %
 8:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 9:
10:  (de windows nil
11:     (fensterinhalt
12:        'windows
```

```
13:        (append
14:           bakliste
15:           '("menüeintrag" "neues Fenster")))
16:     (fenstertitel 'windows " * PoGO-Windows * ")
17:     (fenstermobil 'windows 'off)
18:     (fensterflexibel 'windows 'off)
19:     (fenstervorspann 'windows '(prog ()
20:                      (fensterkoordinaten 'windows
                          'resize)
21:                        (fensterinhalt 'windows
22:                                       (append
23:                                      bakliste
24:                                       '("menüeintrag
                             " "neues Fenster")))))
25:     (fensteraktiv 'windows)
26:     (fenstermethode
27:        'windows
28:        '(prog (f e)
29:            (cond
30:               ((equal el "menüeintrag")
31:                  (printat 1 1 "Menüeintrag in
                                  welches Fenster?")
32:                  (setq f (read))
33:                  (terpri)
34:                  (printat 1 1 "                ")
35:                  (terpri)
36:                  (printat 1 1 "Wie heisst der neue
                                  Eintrag?")
37:                  (setq e (read))(terpri)
38:                  (printat 1 1 "                ")
39:                  (terpri)
40:                  (cond
41:                     ((member e (fensterinhalt f))
42:                        (fensterinhalt f (remove e
                                 (fensterinhalt f))))
43:                     (t
44:                        (fensterinhalt f (append
                        (fensterinhalt f) (list e))))) )
45:               ((equal el "neues Fenster")
46:                  (printat 1 1 "Wie heisst das neue
                                  Fenster?")
47:                  (setq f (read))
48:                  (setq but (form!_alert 1
49:                              (ltoc "[2][Soll das
                                  Fenster mobil sein?]
                              [na klar|no no never]")))
50:                  (cond ((eq but 1)(fenstermobil
                                                f 'on))
51:                        (t (fenstermobil f 'off)))
52:                  (setq but (form!_alert 1
53:                              (ltoc "[2][Soll die
                                  Größe des Fensters|
                                  flexibel sein?]
                             ['türlich|völlig absurd]")))
54:                  (cond ((eq but 1)(fensterflexibel
                                           f 'on))
55:                        (t (fensterflexibel f 'off)))
56:                  (fensteraktiv f)
57:                  (fensteraktiv 'windows)
58:                  (refresh))
59:               (t (fensteraktiv el)
60:                  (schliessefenster 'windows)
61:                  (refresh)))))
62:     (fenster 'windows))
63:
64:
65:  (fensterkoordinaten 'windows '(300 17 500 150))
```

```
 1:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2:  %                           %
 3:  %          LISTENER         %
 4:  %            PoGo           %
 5:  %      by Thomas Mahler     %
 6:  % (c) MAXON Computer GmbH %
 7:  %                           %
 8:  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 9:
10:  (global '(linie))
11:  (fluid '(key ret))
12:
13:
14:  (de wait (OFFSET init)
15:     (setq init offset)
16:     (prog (wert)
17:  begin
```

→

```
18:         (v!_gtext 1
19:                 (iremainder offset 632)
20:                 (iplus (idifference linie 16)
                    (itimes 16 (iquotient offset 632)))
21:                 (ltoc "< "))
22:         (cond
23:            ((eq (multi) 1)
24:               (cond
25:                  ((eq 8 (iremainder key 256))
26:                     (cond
27:                        ((lessp 0 (length wert))
28:                           (setq wert (reverse (cdr
                                        (reverse wert))))
29:                           (setq offset (idifference
                                            offset 8))))
30:                     (go begin))
31:                  ((eq 13 (iremainder key 256))
32:                     (cond
33:                        ((null wert)
34:                           (go begin))
35:                        (t (while (eq (last wert)'!))
36:                              (setq wert (reverse
                                (cdr (reverse wert)))) )
37:                           (return(compress wert)))))
38:                  (t
39:                     (setq wert (append wert (list
                         (ascii (iremainder key 256)))))
40:                     (vswr!_mode 1 1)
41:                     (v!_gtext 1
42:                              (iremainder offset 632)
43:                               (iplus (idifference
                                 linie 16) (itimes 16
                                 (iquotient offset
                                  632)))
44:                               (ltoc(ascii(iremainder
                                        key 256))))
45:                     (setq offset (plus offset 8))
46:                     (go begin))))
47:            (t (return 'fin)))) )
48:
49:
50:  (fluid '(liste))
51:
52:  (de listener nil
53:     (setq liste '("zur Eingabe in den Listener
                 'input :' anklicken !!" input! !:))
54:     (fensterinhalt 'listener liste)
55:     (fensterkoordinaten 'listener '(2 320 638 398))
56:     (fenstertitel 'listener         LISP-LISTENER
                 Mausklick beendet den input-modus")
57:     (fenstermethode 'listener '(listen))
58:     (fensterhistory 'listener 't)
59:     (fensteraktiv 'listener)
```

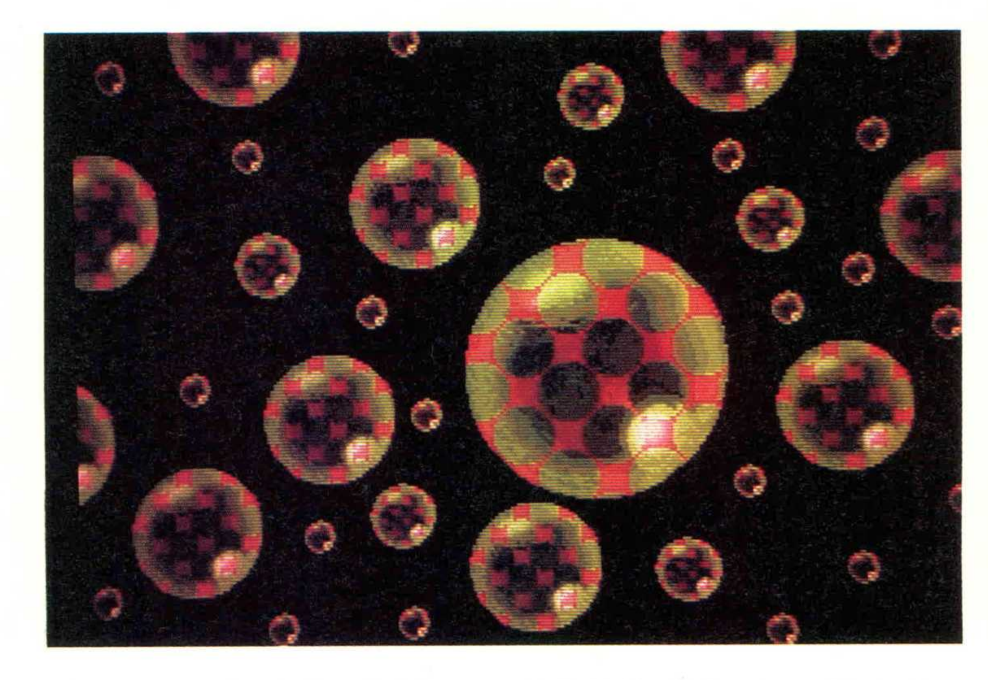
→

```
60:     (fenster 'listener))
61:
62:  (de listen nil
63:     (prog (value input )
64:         (printat 1 1 "")
65:         (fensterkoordinaten 'listener '
                                  (2 50 638 398))
66:         (fenster 'listener)
67:       looping
68:         (setq input (wait 64))
69:         (cond
70:            ((eq input 'fin)
71:               (return (prog ()
72:                          (fensterkoordinaten
                              'listener '(2 320 638
                                        398))
73:                          (refresh))))
74:            (t (setq liste (reverse (rplaca (reverse
                                    liste) input)))
75:               (setq value (errorset  input 0))
76:               (prog (zw)
77:                  (setq liste (append liste '(,(cond
                          ((consp value)(car value))
                          (t '(fehlernr!. ,value)))
80:                     input! !:) ))
81:                loop
82:                  (cond ((lessp (iquotient
                     (idifference (nth 3
                     (fensterkoordinaten 'listener))
83:                          (nth 1 (fensterkoordinaten
                              'listener)))
84:                                         16)
85:                              (add1 (length liste)))
86:                        (setq liste (cdr liste))
                           (go loop)))
87:                  (fensterinhalt 'listener liste)
88:                  (cond ((atom (errorset '(fenster
                                          'listener)0))
89:                        (setq liste '("Dieses
                            Ergebnis lässt sich nicht
                            darstellen !" input! !:))
90:                        (fensterinhalt 'listener
                            liste)
91:                        (fenster 'listener)))))
92:         (go looping) ))
93:
94:  (setq !*comp nil)
95:
96:  (de multi ()
97:     (evnt!_multi 3 2 3 1 0 0 0 0 0 0 0 0 0 0 0 3 4
        mx my mbut stat key ret))
98:  fin
```

Zeit ist Geld

Assembleroptimierung

Das altbekannte Sprichwort "Zeit ist Geld" haben sich diejenigen zu Herzen genommen, die in Assembler programmieren. Das soll natürlich nicht heißen, daß es unmöglich ist, in anderen Sprachen schnelle Programme zu schreiben. Vor allen Dingen in C kann man bei der Geschwindigkeit oft nur staunen. Wenn es jedoch darum geht, ein Maximum an Geschwindigkeit bei gleichzeitig möglichst kompaktem Programmcode zu erreichen, gibt es für die Programmierung in Assembler keine Alternative. Nun ist Assembler jedoch nicht gleich Assembler. Durch geschickte Programmierung unter Ausnutzung spezieller Eigenschaften des 68000-Prozessors lassen sich Assemblerprogramme meist hinsichtlich Programmlänge und Geschwindigkeit noch weiter optimieren. Schauen wir uns einmal an, welche Möglichkeiten der 68000 hier bietet.



Die wohl bekannteste Befehlsgruppe, die sich durch eine schnelle Ausführungszeit bemerkbar macht, stellen die QUICK-Befehle, wie *MOVEQ*, *ADDQ* und *SUBQ* dar. Sie zeichnen sich außerdem dadurch aus, daß die Konstante, die addiert, subtrahiert oder in das angesprochene Register übertragen wird, bereits im Opcode integriert ist, so daß jeder dieser Befehle in nur einem Wort codiert werden kann. Daraus resultieren auch die Einschränkungen bezüglich der Größe der Konstanten, die bei MOVEQ einen Wert von -1 bis -128 bzw. +1 bis +127 und bei SUBQ und ADDQ einen Wert von 1 bis 8 besitzen darf. Der MOVEQ-Befehl darf übrigens nur auf Datenregister angewandt werden. Außerdem ist zu beachten, daß bei ihm grundsätzlich das Datenregister in seiner ganzen Breite, also volle 32 Bit, verändert wird! Wird die obere Hälfte eines Datenregisters noch für spätere Operationen benötigt, darf der MOVEQ-Befehl also nicht verwendet werden, und man muß auf einen "normalen" MOVE.W-Befehl zurückgreifen.

Wie kann man den MOVEQ-Befehl noch anderweitig nutzen? Hier bietet sich das Ersetzen des *CLR.L Dn*-Befehls (Dn steht für ein beliebiges Datenregister) durch ein *MOVEQ.L #0,Dn* an. Beide Befehle erzielen zwar das gleich Ergebnis, aber daß an dieser Stelle die Verwendung eines MOVEQ-Befehle dennoch durchaus sinnvoll sein kann, wird deutlich, wenn man sich die Ausführungszeit für den CLR.L Dn-Befehl ansieht. Diese beträgt nämlich 8 Taktzyklen. MOVEQ.L #0,Dn benötigt dagegen nur 4 Zyklen, also die halbe Ausführungszeit bei gleicher Anzahl an Bytes.

Auch bei der Optimierung des folgenden Befehls ist der MOVEQ-Befehl wertvoll:

```
MOVE.L #1,-(SP)
```

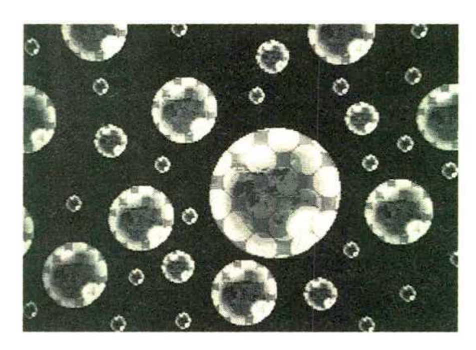
Dieser Befehl kommt bei Aufrufen des Betriebssystems auf dem ATARI ST recht häufig vor. Oft muß hier als Parameter -1 übergeben werden, wenn man einen Parameter vom Betriebssystem erfragen

will. Ein Beispiel stellt die MALLOC-Routine dar, bei der ein Langwort mit dem Wert -1 auf den Stack übergeben werden muß, um den noch freien Speicherplatz zu ermitteln. Der Speicherbedarf für den obigen Befehl beträgt 6 Bytes, die Ausführungszeit 20 Taktzyklen. Falls man ein Datenregister frei hat, kann man den obigen Befehl durch die folgende Befehlsfolge ersetzen:

```
MOVEQ.L #1,Dn
MOVE.L Dn,-(SP)
```

Nicht verwendete Register sind vor einem Aufruf des Betriebssystems eigentlich immer vorhanden, da die Inhalte einiger Register beim Aufruf ohnehin überschrieben werden. So kommt D0 für dieses Beispiel auf jeden Fall in Frage. Obwohl im obigen Beispiel nun zwei Befehle vom Prozessor ausgeführt werden müssen, verringert sich der Speicherbedarf auf 4 Bytes bei einer Ausführungszeit von nur noch 18 Taktzyklen. Also hat man wiederum Zeit und zusätzlich noch Speicherplatz gespart (auch wenn dieser beim ST im allgemeinen nicht zu knapp ist).

Verlassen wir die QUICK-Befehle und wenden uns den "langsamen" Befehlen des Prozessors zu. Nach einem Aufruf des Betriebssystem ist es nötig, den Stack von den übergebenen Parametern zu befreien. Bei weniger als 9 Parametern geschieht dies durch ein *ADDQ.L #n,SP*. Dabei stellt *n* eine Konstante von 1 bis 8 dar, wie sie für einen ADDQ-Befehl erlaubt ist. Bei mehr als 8 Parametern findet man oft den Befehl *ADD.L #n,SP*. Benötigt der ADDQ.L-Befehl nur 2 Bytes, so werden durch ADD.L schon 6 Bytes verbraucht, wovon 4 Bytes auf die Konstante *n* entfallen, die ja ein Langwort darstellt. Dabei kann es nach einem Betriebssystemaufruf niemals vorkommen, daß diese Konstante einen Wert besitzt, der von der Größenordnung eines Langworts ist. So betrachtet würde also ein *ADD.W #n,SP* bereits genügen. Der Vorteil liegt in 2 eingesparten Bytes und in 12 statt 14 Taktzyklen für die Ausführungszeit. Vielleicht stellen Sie Sich nun die Frage, ob es denn überhaupt erlaubt ist, Wortverarbeitung zu wählen, da der Stackpointer doch ein 32-Bit-Register darstellt. In diesem Zusammenhang spielt es eine Rolle, daß der 68000-Prozessor bei Additionen und Subtraktionen in Zusammenhang mit einem Adreßregister (also auch beim Stackpointer, der ja durch A7 repräsentiert wird) die Konstante automatisch vorzeichenrichtig auf ein Langwort erweitert, wenn es sich bei dieser Konstante um ein Wort handelt. Aus dem Wort *n* macht der Prozessor also intern ein Langwort *n* und



addiert oder subtrahiert es erst dann vom Adreßregister. Das Ganze funktioniert aber nur, wenn es sich bei dem angesprochenen Register um ein Adreßregister handelt! Addiert man ein Wort in einem Datenregister auf ein Adreßregister, so braucht dies übrigens mehr Zeit, als wenn man mit einem Langwort arbeitet. Dies liegt daran, daß in diesem Fall für die vorzeichenrichtige Erweiterung 2 Taktzyklen mehr benötigt werden.

Um beim Entfernen von Parametern, die sich auf dem Stack befinden, noch effektiver zu sein, kann man den ADD-Befehl durch einen LEA-Befehl ersetzten. *ADD.W #12,SP* wird somit zu *LEA 12(SP),SP*. Dieser Befehl entspricht in seiner Wirkung einer Addition, benötigt im Gegensatz zu dieser aber nur 8 statt 12 Taktzyklen.

Beim MOVE-Befehl verhält sich der Prozessor übrigens ähnlich wie beim ADD-Befehl. Auch hier wird ein Wort in ein Langwort umgewandelt. Der Befehl *MOVE.W #$2000,An* erfüllt also die gleiche Aufgabe wie *MOVE.L #$2000,An* und das mit 2 Bytes weniger an Speicherbedarf und einer um 4 Zyklen geringeren Ausführungszeit. Auch hier wird man allerdings stattdessen normalerweise einen LEA-Befehl verwenden.

Ein Befehl, der nur selten benötigt wird, ist der *MOVEP*-Befehl. Er ist dazu gedacht, den Inhalt eines Datenregisters in hintereinanderfolgende Hardwareadressen zu schreiben oder aus diesen zu lesen. Dies soll ein wenig näher erläutert werden, da der MOVEP-Befehl zu den weniger bekannten Befehlen gehört. Wie Sie vielleicht wissen, sind die Speicherzellen, über die die Hardware des ST angesprochen werden kann, quasi "auf Lücke" angeordnet. Ist *adr* die Basisadresse für irgendeinen Hardwarechip, so liegen die zugehörigen Register auf den Adressen *adr*, *adr+2*, *adr+4* usw. Ein Byte zwischen jeder Adresse ist also ungenutzt und kann vom Prozessor nicht angesprochen werden (ein Busfehler wäre die Folge). Eine praktische Anwendung des MOVEP.W-Befehls beim ST ist das Selektieren und darauffolgende Beschreiben eines Registers des Soundchips. Der MOVEP-Befehl teilt das Datenwort in zwei Bytes auf, von denen das erste an die Basisadresse des Chips $FFFF8800 und das zweite nach $FFFF8800+2 geschrieben wird. (Die Adresse $FFFF8800 sorgt hier für die Registerauswahl des Soundchips, die nächste Adresse ermöglicht einen Zugriff auf das zuvor selektierte Register.) Bei Langworten verhält sich der MOVEP-Befehl ähnlich, hier werden natürlich alle vier Bytes des Datenregisters übertragen.

Im folgenden Beispiel habe ich den MOVEP-Befehl ein wenig zweckentfremdet und verwende ihn dazu, vor AES-Aufrufen die notwendigen Parameter in das CONTROL-Array des AES zu übertragen. Wie das geschieht, kann man Listing 1 entnehmen.

Die vier Parameter *b1 bis b4* stellen ebenso wie die Funktionsnummern normalerweise Bytewerte dar und können so problemlos in einem Langwort untergebracht werden. Dieses Langwort kann dann in einem Schritt mit dem MOVEP-Befehl ins CONTROL-Array übertragen werden. Meiner Meinung nach stellt dies die schnellste Möglichkeit dar, Parameter an das AES zu übergeben. Darüber hinaus ist diese Vorgehensweise recht komfortabel.

```
MOVEQ #OPCODE,D0          ;Nummer der gewünschten AES-Funktion
MOVE.L #$b1b2b3b4,D1      ;Parameter für das CONTROL-Array
LEA CONTRL,A0             ;Adresse des CONTROL-Arrays
MOVE D0,(A0)              ;Funktionsnummer eintragen
MOVEP.L D1,2(A0)          ;alle Parameter auf einmal übertragen
```

Listing 1: Übertragen der notwendigen Parameter vor einem AES-Aufruf

Bei der Programmierung in Assembler sollte man sich bei zeitkritischen Programmteilen stets bewußt sein, daß ein direkter Zugriff auf den Speicher zeitlich ungünstiger ist, als wenn man möglichst ausgiebig mit den Adreßregistern arbeitet.

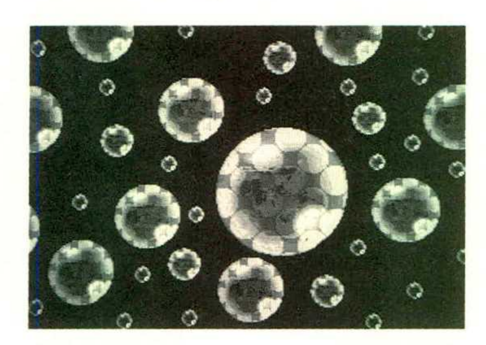
```
MOVE.L #KONSTANTE1,adresse
MOVE.L #KONSTANTE2,adresse+2
```

Schon bei zwei Zugriffen auf den Speicher, wie im obigen Beispiel, ist es, wenn man die Ausführungszeit betrachtet, günstiger, den Umweg über ein Adreßregister zu gehen (natülich unter der Voraussetzung, es ist noch ein freies vorhanden):

```
LEA adresse,A0
MOVE.L #KONSTANTE1,(A0)+
MOVE.L #KONSTANTE2,(A0)
```

Diese Variante erfüllt den gleichen Zweck wie das erste Beispiel, benötigt allerdings nur 52 statt 56 Taktzyklen. Sollen mehr als zwei Daten auf diese Weise in den Speicher geschrieben werden, so verbessert sich dieses Verhältnis noch, auch bezüglich des Speicherplatzbedarfs.

Eine Beschleunigung von Programmen wird nicht nur durch eine sinnvolle Anwendung einzelner Befehle sondern auch durch die richtige Wahl der Adressierungsart erreicht. Als besonders praktisch erweist sich hier die programmzählerrelative Adressierung. Beim Ersetzen einer absoluten Adresse durch eine programmzählerrelative Adressierung werden zwei Bytes Speicherplatz und natürlich auch Rechenzeit gespart. Leider kann diese Art der Adressierung nicht bei jedem Be-



fehlstyp verwendet werden. Insbesondere Schreibzugriffe sind programmzählerrelativ nicht erlaubt. Weiter muß beachtet werden, daß sich die Adresse, auf die man sich bei der programmzählerrelativen Adressierung bezieht, nicht weiter als 32k von der Position des Befehls, der diese Adressierung beinhaltet, entfernt sein darf. Bei der PC-relativen Adressierung gibt man nämlich nicht die absolute Adresse an, auf die zugegriffen werden soll sondern nur die relative Entfernung dieser Adresse zum Programmzähler. Dieser Offset wird in einem Wort abgelegt, daher die Beschränkung auf 32k-Entfernung des PCs für diese Adressierung. Als Programmierer braucht man sich jedoch nicht mit der Berechnung des Adreßoffsets herumzuschlagen, das übernimmt der Assembler.

Es ist nicht möglich, auf Daten im DATA-Segment eines Programms über die programmzählerrelative Adressierung zuzugreifen. Diese Einschränkung beruht darauf, daß erst nach dem Linken eines Programms feststeht, ob die durch diese Adressierung angesprochenen Adressen im Speicher nicht weiter als 32k vom zugehörigen Befehl entfernt sind. Zu diesem Zeitpunkt ist es jedoch nicht mehr möglich, Einfluß auf eine Adressierungsart zu nehmen. Man kann jedoch bei entsprechender Programmierung sicherstellen, daß man auch auf Teile des Datenbereichs programmzählerrelativ zugreifen kann, auch wenn diese Art der Programmierung nicht ganz sauber ist. Es genügt nämlich, alle Daten, auf die programmzählerrelativ zugegriffen werden soll, nicht im DATA- sondern im TEXT-Segment unterzubringen. Die Unterscheidung zwischen TEXT- und DATA-Segmenten ist ja schließlich nur eine logische Trennung, die weiter keine Bedeutung besitzt. Dabei muß sich jedoch darüber im Klaren sein, daß man so die an sich sinnvolle Unterteilung eines Assemblerprogramms in TEXT-, DATA- und BSS-Segment untergräbt. Hier gilt das Motto: Der Zweck heiligt die Mittel.

Man kann sogar noch einen Schritt weitergehen, indem man auch Definitionen, die eigentlich ins BSS-Segment gehören, im TEXT-Segment des Programms unterbringt. Dies erweist sich dann als günstig, wenn auf bestimmte Variablen des BSS-Segements häufige Lesezugriffe erfolgen wie im folgenden Fall:

```
MOVE.B adr1,D0
  .
MOVE.B adr1,adr2
  .
  .
```

Ist die Variable *adr1* im BSS-Segment definiert, so benötigt diese Befehlssequenz 16 Bytes und 44 Taktzyklen. Definieren Sie diese Variable im TEXT-Segment, so daß Sie programmzählerrelativ darauf zugreifen können, so werden nur noch 12 Bytes und 36 Takzyklen benötigt. Einen Haken hat die Sache jedoch, der den aufmerksamen Leser vielleicht schon stutzig gemacht hat: Werden Variablen, die eigentlich im BSS-Teil untergebracht wären, im TEXT-Segment definiert, so schlägt sich dies in der Programmlänge nieder, da das gesamte TEXT-Segment sich auf der Diskette befindet. Befinden sich Variablen im BSS-Segment, so tritt dieser Umstand nicht auf, da hier nur eine Information über die gesamte Länge dieses Segmentes auf der Diskette vorhanden ist. Wenn man nun in diese Richtung weiterdenkt, ist jedoch logisch, daß, wenn sich ein Programm einmal im Speicher des ST befindet, es keinen Unterschied mehr macht, welche Variable in welchem Segment definiert worden ist, da nun auf jeden Fall der benötigte Speicherplatz vom Betriebssystem zur Verfügung gestellt werden muß. Fazit: Werden Variablen vom BSS- ins TEXT-Segment verlegt, so steigt der Platzbedarf des Programms auf Diskette um so viele Bytes, wie für diese Variablen reserviert wird. Da aber nun ein programmzählerrelativer Zugriff auf diese Variablen möglich ist, wird in unserem Beispiel schon bei zwei Zugriffen der Speicherplatzverlust wettgemacht.

```
MOVE.B adr1(PC),D0
  .
  .
MOVE.B adr1(PC),adr2
  .
  .
```

Hier wird zweimal die programmzählerrelative Adressierung verwendet, was jeweils zwei Bytes weniger ausmacht als ein direkter Zugriff. Die folgende Rechnung können wir auch ohne Computer im Kopf durchführen: 2 Bytes mehr auf Diskette, 4 Bytes durch Verwendung einer anderen Adressierungsart eingespart = 2 Bytes insgesamt auf Diskette gespart.

Zusätzlich kann sich noch die verkürzte Ausführungszeit positiv auswirken.

Am Abschluß dieses Artikels möchte ich noch auf einen speziellen Befehl zu sprechen kommen, der meiner Meinung nach zu selten eingesetzt wird, aber manche Befehlssequenz der folgenden Form vereinfachen kann:

```
      TST.B adr
      BEQ lbl
      .
      .
lbl:  ST adr
      .
      .
```

adr stellt hier also ein Flag dar, dessen Inhalt für den weiteren Verlauf des Programms ausschlaggebend ist. Wenn Sie sich überlegen, welchen Inhalt die Variable *adr* nach der obigen Befehlssequenz besitzt, so fällt auf, daß sie auf jeden Fall einen Wert ungleich Null besitzt, egal welcher Programmzweig abgearbeitet wird. Dies ist aber auf folgdendem Weg schneller und prägnanter zu erreichen:

```
      TAS adr
      BEQ lbl
      .
      .
lbl:  .
      .
```

Durch den TAS-Befehl wird das Flag (Semaphor) *adr* gesetzt und gleichzeitig wird das Statusregister entsprechend dem ursprünglichen Inhalt von *adr* verändert. Durch diese Programmierung sind also ein TST- und ein ST-Befehl im TAS-Befehl zusammengefaßt worden.

In den diversen Beispielen ist sicher deutlich geworden, daß bei gut durchdachter Programmierung auch die Geschwindigkeit von Assemblerroutinen sowie deren Speicherplatzbedarf optimiert werden kann. Manchem Leser bin ich mit der Optimierung sicherlich ein wenig zu weit gegangen, da die Programmierung an vielen Stellen nicht mehr ganz sauber war. Man bedenke jedoch, daß gerade Interrupt- oder manche Ein-/Ausgabe-Routinen nur ein Minimum an Zeit benötigen sollten, so daß hier oft nach ungewöhnlichen Verfahren für eine Optimierung gesucht werden muß. Ich habe die Erfahrung gemacht, daß sich fast jede Routine noch optimieren läßt, wenn man sie nur scharf und lange genug durchsieht. Wichtig ist natürlich, daß man Informationen über Länge und Ausführungszeit der einzelnen Befehle besitzt. Solche Angaben können z.B. dem Buch "M68000-Familie", Band I, von Werner Hilf und Anton Nausch entnommen werden.

Uwe Seimet



Der leuchtende Pfad

Interface für HP-Taschenrechner

Manche Studenten, die zu Hause einen ATARI ST auf dem Schreibtisch stehen haben, möchten diesen gerne in die Aula mitnehmen. Das nicht geringe Volumen des genannten Rechners macht jedoch jede Hoffnung zunichte. Bis einer der Studenten sich mit der Idee vertraut gemacht hat, Stacy nur in der Absicht zu kaufen, in einer schwierigen Klausur die Berechnung auf dem ST laufen zu lassen, werden mit Sicherheit Jahre vergehen, und der Student wird keiner mehr sein.

Es geht aber einfacher. Man muß sich nicht unbedingt mit einem 16 Bit-Rechner bei Klausuren helfen, es reicht auch ein guter Taschenrechner. Und die sind bekanntlich sehr klein. Es gibt sie in jeder Form und in fast jeder Farbe. Einige können mehr als das, was ein Student irgendwann lernen wird, andere allerdings haben viele Tasten, sind aber unbrauchbar. Unter diesen vielen Taschenrechnern befindet sich der brauchbare, aber teure Hewlett Packard, der seit mehreren Jahren in verschiedenen Ausführungen eine der führenden Rollen auf diesem Gebiet spielt. Seit kurzem wird ein Interface angeboten, das die Kommunikation von HP-Taschenrechnern der gehobenen Klasse (HP-17B, HP-19B, HP-27S, HP-28C, HP-28S und alle Rechner der 41er Serie) mit dem ATARI ST ermöglicht.

Kabelloser Kontakt

Die Kommunikation zwischen HP-Taschenrechner und ATARI ST erfolgt über eine eingebaute Schnittstelle beim HP-Rechner (beim HP-41 ist diese Schnittstelle nicht eingebaut, man muß sie separat kaufen) und einem Interface, das an der seriellen Schnittstelle des ATARI eingesteckt wird. Diese Schnittstelle des HP-Rechners bildet das Bindeglied zwischen Taschenrechner und Drucker und wird über eine Infrarot-Diode realisiert. Auf der anderen Seite (der ST-Seite) empfängt das Interface die optischen Daten, die vom Taschenrechner gesendet werden, und bereitet die Daten so auf, daß der ST damit etwas anfangen kann. Ist zwischen beiden Rechnern freie Sicht und beträgt die Entfernung nicht mehr als 1,5 Meter, ist fast immer eine fehlerlose Übertragung gesichert. Aber mit dem Interface ist nur die Hälfte der notwendigen Arbeit geleistet, einen großen Teil übernimmt die mitgelieferte Software.

Weiche Übertragung

HP-Transfer heißt das Programm, das die Daten von einem Rechner zum anderen überträgt. Die Software ist vollständig in GEM eingebunden, und die Bedienung erfolgt über die gewohnten Pull-Down-Menüs. Man kann mit Hilfe der anwählbaren Menüs Daten empfangen, speichern, weiterverarbeiten, abdrucken etc. Für alle diese Funktionen werden zahlreiche Unterfunktionen angeboten, so daß die Einarbeitung in dieses Programm sehr leicht fällt.

Eins der wichtigsten Anwendungsgebiete dieser Übertragung ist die Möglichkeit, Grafik, die mit dem HP-Taschenrechner erzeugt wird, in einer besseren Qualität auf dem ST auszudrucken. Dafür stellt die Software die Möglichkeit, ihre eigene Druckerkonfiguration von Diskette einzuladen. Ebenfalls besteht durch das mitgelieferte Programm die Möglichkeit, ganze Bildschirme in verschiedenen Formaten zu speichern.

Wie oben erwähnt ist dieses Interface für jene Anwender von großem Interesse, die beide Rechner besitzen, und die Daten des Taschenrechners irgendwie weiterverarbeiten müssen. Es sollte auch gesagt werden, daß die übertragung nur vom HP-Rechner zum ATARI erfolgt, so daß es nicht möglich ist, Programme auf dem ST zu entwickeln (mit einer größeren Tastatur und einem besseren Sichtgerät) und auf einen HP-Rechner zu übertragen. Vielleicht ergibt sich in Zukunft eine solche Alternative, aber in der Zwischenzeit müssen wir uns mit einer leuchtenden Sackgasse zufriedengeben. Das Interface und die Software kosten zusammen DM 280,-.

Bezugsadresse:

Kai-Uwe Packheiser
Rechtstraat 39
Valmeer/Belgien
Tel.: 0032/12/45617

Druckerunabhängige Ausdrucke in OMIKRON.BASIC

Wer kennt das nicht? Da hat man nun das absolute Programm geschrieben, will es veröffentlichen und muß dann feststellen, daß die Ausdruckroutinen, die man eingebaut hat, zwar auch auf einigen anderen Druckern laufen, nicht aber auf all denen, die im Handel und weitverbreitet sind.

GST hat dies bei seinem Textverarbeitungssystem 1st Word Plus berücksichtigt und erlaubt, die wichtigsten Steuerroutinen für den Drucker aus einem Treiber zu entnehmen. Und da 1st Word Plus weitverbreitet ist, wurden jede Menge von geeigneten Treibern für fast alle am Markt verbreiteten Drucker geschrieben.

Ist man nun bereit, sich bei seinen Programmen auf die in den *.CFG Dateien vorhandenen Routinen zu beschränken, kann man diese problemlos auch in OMIKRON.BASIC nutzen. Die in Tabelle 1 vorliegende Sammlung von Prozeduren ermöglicht die Nutzung folgender Druckerfunktionen. Diese Prozeduren berücksichtigen die Eigenschaft von 1st Word Plus, eine NLQ-Steuerfunktion, falls diese undefiniert ist, durch die entsprechende Draft-Funktion zu ersetzen.

Bei der Benutzung der Steuerfunktionen muß beachtet werden, daß die Umschaltung NLQ/Draft in der Regel durch die Lineale und nicht durch die Attribut-Umschaltungen geschieht.

Ans "Eingemachte"

Im folgenden wird das Listing dokumentiert: Zunächst werden zwei benötigte Arrays dimensioniert und die Install-Prozedur aufgerufen.
Anschließend folgt die eigentliche *.CFG-Installationsroutine. Als Vorgabe werden in das Array Conv$(I%L) die Werte eingesetzt, die ohne Umwandlung ausgegeben würden.

Als nächstes wird die gewünschte *.CFG-Datei ausgewählt. Dabei wird überprüft, ob überhaupt eine Datei ausgewählt wurde. Ist dies der Fall, wird sie geöffnet, in den String Steuer_String$ übertragen und wieder geschlossen.

Fett_Ein	Sub_Ein
Fett_Aus	Sub_Aus
Nlq_Fett_Ein	Nlq_Sub_Ein
Nlq_Fett_Aus	Nlq_Sub_Aus
Kurs_Ein	Unter_Ein
Kurs_Aus	Unter_Aus
Nlq_Kurs_Ein	Nlq_Unter_Ein
Nlq_Kurs_Aus	Nlq_Unter_Aus
Hell_Ein	Norm_6_Cpi
Hell_Aus	Norm_10_Cpi
Nlq_Hell_Ein	Norm_12_Cpi
Nlq_Hell_Aus	Norm_17_Cpi
Super_Ein	Nlq_6_Cpi
Super_Aus	Nlq_10_Cpi
NLQ_Super_Ein	Nlq_12_Cpi
Nlq_Super_Aus	Nlq_17_Cpi

Die Druckfunktionen von 1st Word Plus

Danach wird überprüft, ob es sich bei der geöffneten Datei um einen 1st Word Plus *.CFG-Treiber handelt. Falls dies der Fall ist, wird der Name des gewählten Druckertreibers angezeigt.

Jetzt werden nacheinander zuerst die verschiedenen Steuerfunktionen des Treibers in das Array Func$(I) übertragen, anschließend kommt die Übersetzungstabelle für die Sonderzeichen an die Reihe. Sie werden in das Array Conv$(I) übertragen. Ist für ein Zeichen kein Eintrag vorhanden, wird es nicht verändert. (Conv$(I) = I { siehe Zeile 9 })

Es folgen die Prozeduren, die dann im Programm aufgerufen werden müssen, um den gewünschten Effekt zu erzielen. Aus Platzgründen habe ich diejenigen, die keine IF-Abfrage enthalten, in eine Zeile gesetzt, obwohl es natürlich "schöner" gewesen wäre, sich auf einen Befehl pro Zeile zu beschränken. Aber so werden einige hundert Bytes eingespart.

Am Ende kommt noch eine Funktion, die es erlaubt, die auszudruckenden Zeichen entsprechend der Übersetzungstabelle des *.CFG-Treibers umzuwandeln. Auch Zeichen, die durch Einzelnadelgrafik erzeugt werden, sind so darstellbar.

Klaus Opel

```
0 '
1 DIM Func$(51),Conv$(255)  ' Array-Dimensions.
2 '
3 Install                   ' Prozeduraufruf
4 '
5 STOP
6 '
7 DEF PROC Install          ' CFG-Installations-
                              Routine
8   FOR I%L=0 TO 255        ' Vorgabewerte
9     Conv$(I%L)= CHR$(I%L)
10    NEXT I%L
11    Steuer_String$=""
12    CLS
13    Pfad$="A:\*.CFG"       ' Dateiauswahl
14    MOUSEON
15    FILESELECT (Pfad$,File$,Knopf%L)
16    MOUSEOFF
17 '
18    IF Knopf%L<>0 AND File$<>"" THEN
19      Datei$= LEFT$(Pfad$,( LEN(Pfad$)-5))+File$
                                  ' Datei vorhanden
20      CLS
21      OPEN "R",1,Datei$,1
22      FIELD 1,1 AS Zeichen$
23      FOR I%L=1 TO LOF(1)
24        GET 1,I%L
25        Steuer_String$=Steuer_String$+Zeichen$
26      NEXT I%L    ' Übertragung in Steuer_String$
27      CLOSE 1
28 '
29      IF LEFT$(Steuer_String$,8)="GST-CFG:" THEN
                                ' Datei = CFG-Datei?
30        Steuer_String$= MID$(Steuer_String$,9)
31        Name_End%L=INSTR(Steuer_String$, CHR$(0))
32        Inst_Name$= LEFT$(Steuer_String$,
                         Name_End%L)
33        PRINT @(2,13);"Der von Ihnen angewählte
                   Druckertreiber hat den Namen"
34        PRINT @(6,(80-Name_End%L)/2);;Inst_Name$
35        Steuer_String$= MID$(Steuer_String$,
                            Name_End%L+1)
36        Steuer_String$= MID$(Steuer_String$,7)
37 '
38        WHILE ASC(Steuer_String$)<>0
      ' Übertragen der Steuerfunktionen in Func$(I)
39          Func_Len%L= ASC(Steuer_String$)
40          Func_Num%L= ASC( MID$(Steuer_String$,2,
                           1))
41          Func$= LEFT$(Steuer_String$,Func_Len%L)
42          Func$(Func_Num%L)= RIGHT$(Func$,
                              Func_Len%L-2)
43          Steuer_String$= MID$(Steuer_String$,
                              Func_Len%L+1)
44        WEND
45        Steuer_String$= MID$(Steuer_String$,2)
46 '
47        WHILE ASC(Steuer_String$)<>0
       ' Übertragen der Übersetzungstabelle
         f. Sonderzeichen
48          Conv_Len%L= ASC(Steuer_String$)
49          Conv_Num%L= ASC( MID$(Steuer_String$,2,
                           1))
50          Conv$= LEFT$(Steuer_String$,Conv_Len%L)
51          IF Conv_Len%L>2 THEN
52            Conv$(Conv_Num%L)= RIGHT$(Conv$,
                                  Conv_Len%L-2)
53          ELSE
54            Conv$(Conv_Num%L)=" "
55          ENDIF
56          Steuer_String$= MID$(Steuer_String$,
                              Conv_Len%L+1)
57        WEND
58      ELSE    ' Fehlerausgaben
59        PRINT @(2,2);" Sie haben keinen 1st Word
              Plus Druckertreiber angewählt !!!"
60      ENDIF
61    ELSE
62      CLS
63      PRINT @(10,10);"Es wurde keine Datei
                  angewählt !!!"
64    ENDIF
65 '
66 RETURN
67 '
68 'Druckersteuerungsprozeduren
69 'Word Plus 06
70 DEF PROC Fett_Ein: LPRINT Func$(6);: RETURN
71 '
72 'Word Plus 07
73 DEF PROC Fett_Aus: LPRINT Func$(7);: RETURN
74 '
75 'Word Plus 08
76 DEF PROC Nlq_Fett_Ein
77   IF Func$(8)="" THEN Fett_Ein ELSE LPRINT
                        Func$(8);
78 RETURN
79 '
80 'Word Plus 09
81 DEF PROC Nlq_Fett_Aus
82   IF Func$(9)="" THEN Fett_Aus ELSE LPRINT
                        Func$(9);
83 RETURN
84 '
85 'Word Plus 0A
86 DEF PROC Kurs_Ein: LPRINT Func$(10);: RETURN
87 '
88 'Word Plus 0B
89  DEF PROC Kurs_Aus: LPRINT Func$(11);: RETURN
90 '
91 'Word Plus 0C
92 DEF PROC Nlq_Kurs_Ein
93   IF Func$(12)="" THEN Kurs_Ein ELSE LPRINT
                         Func$(12);
94 RETURN
95 '
96 'Word Plus 0D
97 DEF PROC Nlq_Kurs_Aus
98   IF Func$(13)="" THEN Kurs_Aus ELSE LPRINT
                         Func$(13);
99 RETURN
100 '
101 'Word Plus 0E
102 DEF PROC Hell_Ein: LPRINT Func$(14);: RETURN
103 '
104 'Word Plus 0F
105 DEF PROC Hell_Aus: LPRINT Func$(15);: RETURN
106 '
107 'Word Plus 10
108 DEF PROC Nlq_Hell_Ein
109   IF Func$(16)="" THEN Hell_Ein ELSE LPRINT
                          Func$(16);
110 RETURN
111 '
112 'Word Plus 11
113 DEF PROC Nlq_Hell_Aus
114   IF Func$(17)="" THEN Hell_Aus ELSE LPRINT
                          Func$(17);
115 RETURN
116 '
117 'Word Plus 12
118 DEF PROC Super_Ein: LPRINT Func$(18);: RETURN
119 '
120 'Word Plus 13
121 DEF PROC Super_Aus: LPRINT Func$(19);: RETURN
122 '
123 'Word Plus 14
124 DEF PROC Nlq_Super_Ein
125   IF Func$(20)="" THEN Super_Ein ELSE LPRINT
                          Func$(20);
126 RETURN
127 '
128 'Word Plus 15
129 DEF PROC Nlq_Super_Aus
130   IF Func$(21)="" THEN Super_Aus ELSE LPRINT
                          Func$(21);
131 RETURN
132 '
133 'Word Plus 16
134 DEF PROC Sub_Ein: LPRINT Func$(22);: RETURN
135 '
136 'Word Plus 17
137 DEF PROC Sub_Aus: LPRINT Func$(23);: RETURN
138 '
139 'Word Plus 18
140 DEF PROC Nlq_Sub_Ein
141   IF Func$(24)="" THEN Sub_Ein ELSE LPRINT
                          Func$(24);
142 RETURN
143 '
144 'Word Plus 19
145 DEF PROC Nlq_Sub_Aus'
146   IF Func$(25)="" THEN Sub_Aus ELSE LPRINT
                          Func$(25);
```

→

```
147 RETURN
148 '
149 'Word Plus 1A
150 DEF PROC Unter_Ein: LPRINT Func$(26);: RETURN
151 '
152 'Word Plus 1B
153 DEF PROC Unter_Aus: LPRINT Func$(27);: RETURN
154 '
155 'Word Plus 1C
156 DEF PROC Nlq_Unter_Ein
157   IF Func$(28)="" THEN Unter_Ein ELSE LPRINT
                          Func$(28);
158 RETURN
159 '
160 'Word Plus 1D
161 DEF PROC Nlq_Unter_Aus
162   IF Func$(29)="" THEN Unter_Aus ELSE LPRINT
                          Func$(29);
163 RETURN
164 '
165 'Word Plus 28
166 DEF PROC Norm_10_Cpi: LPRINT Func$(40);:
    RETURN
167 '
168 'Word Plus 29
169 DEF PROC Nlq_10_Cpi
170   IF Func$(41)="" THEN Norm_10_Cpi ELSE LPRINT
                          Func$(41);
171 RETURN
172 '
173 'Word Plus 2A
174 DEF PROC Norm_12_Cpi: LPRINT Func$(42);:
    RETURN
175 '
176 'Word Plus 2B
177 DEF PROC Nlq_12_Cpi
178   IF Func$(43)="" THEN Norm_12_Cpi ELSE LPRINT
                          Func$(43);
179 RETURN
180 '
181 'Word Plus 2C
182 DEF PROC Norm_17_Cpi: LPRINT Func$(44);:
    RETURN
183 '
184 'Word Plus 2D
185 DEF PROC Nlq_17_Cpi
186   IF Func$(45)="" THEN Norm_17_Cpi ELSE LPRINT
                          Func$(45);
187 RETURN
188 '
189 'Word Plus 2E
190 DEF PROC Norm_6_Cpi: LPRINT Func$(46);: RETURN
191 '
192 'Word Plus 2F
193 DEF PROC Nlq_6_Cpi
194   IF Func$(47)="" THEN Norm_6_Cpi ELSE LPRINT
                          Func$(47);
195 RETURN
196 '
197 '
198 DEF FN Convert$(Xyz$) ' Umwandlung der
                          Übersetzungstabelle
199   Zyx$=""
200   FOR I%L=1 TO LEN(Xyz$)
201     Zyx$=Zyx$+Conv$( ASC( MID$(Xyz$,I%L,1)))
202   NEXT I%L
203 RETURN Zyx$
204 END
```

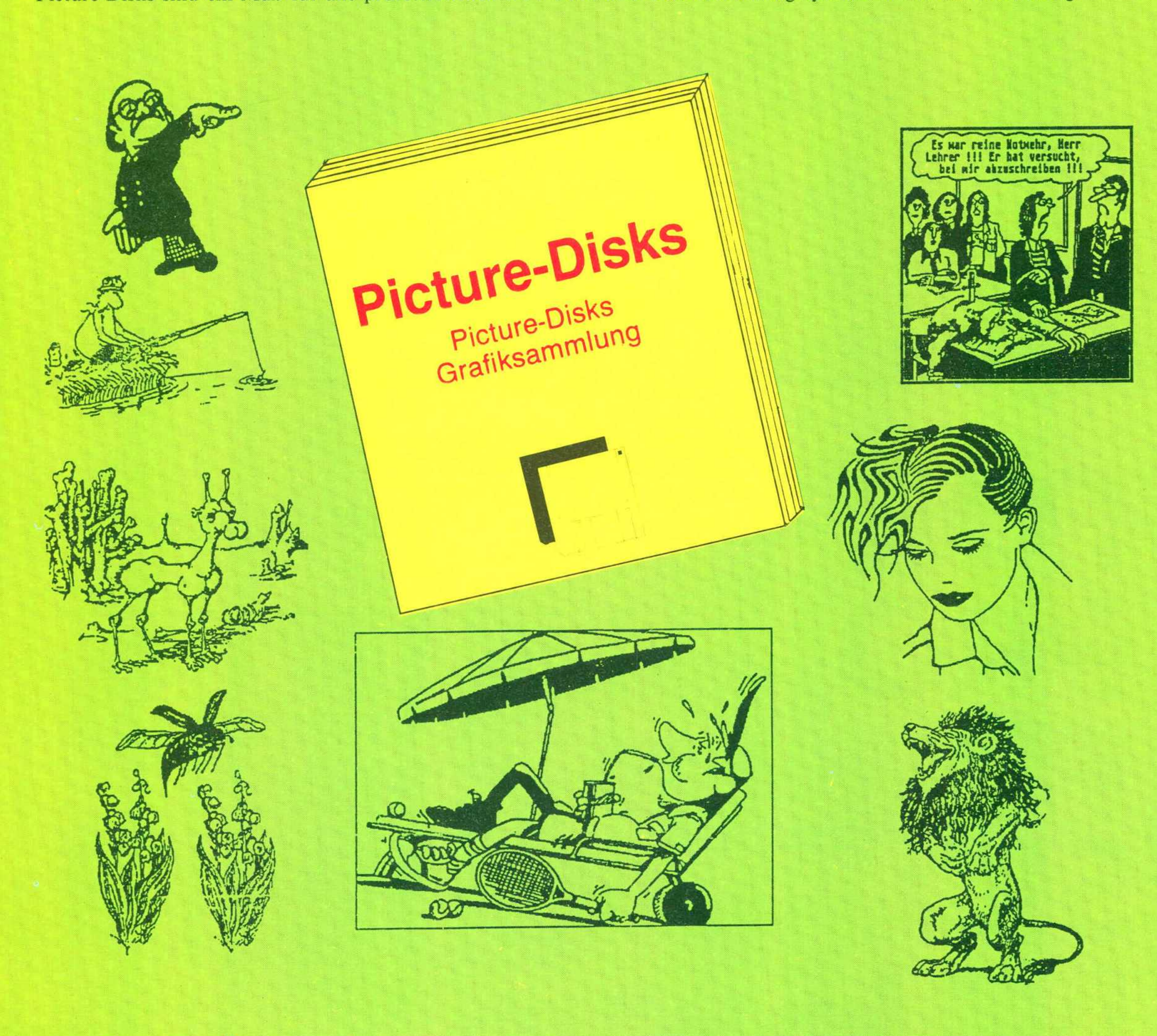
Jetzt für ATARI

PICTURE-DISKS GRAFIKSAMMLUNG

Die Picture-Disks-Serie ist die erste Graphik-Sammlung mit hochauflösenden Graphiken für ATARI ST Computer. Die Picture-Disks sind von der Qualität einzigartig und decken alle Themenbereiche ab.

1000 Graphiken DM 129,—
2000 Graphiken DM 249,—

Die Grafiken können direkt in das Programm STAD oder SIGNUM II eingelesen werden. Für andere Programme steht ein Accessory zur Verfügung, das die Grafiken in DEGAS- oder DOODLE-Format übersetzt. Damit steht für jeden Anwendungsbereich das passende Format zur Verfügung. Den Picture-Disks liegt jeweils ein Handbuch bei, das Ihnen ein schnelles Auffinden und Arbeiten mit den Picture-Disks ermöglicht.
Picture-Disks sind ein Muß für alle professionellen DTP Mal- und Textverarbeitungssysteme mit Grafikeinbindung.



Heim Verlag

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

Schweiz: DataTrade AG
Langstr. 94
CH-8021 Zürich

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

* alle Preise sind unverbindlich empfohlene Verkaufspreise

Ich bestelle
☐ 1000 PICTURE-DISKS zum Preis von DM 129,—
☐ 2000 PICTURE-DISKS zum Preis von DM 249,—
zzgl. DM 5,– Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnr. ____________________

PLZ, Ort ____________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

ALMO Statistiksystem

Programme, die sich mit Mathematik beschäftigen, gibt es für den ST nicht gerade wie Sand am Meer. Hier soll das Statistikprogramm ALMO vorgestellt werden. ALMO steht für "Allgemeines lineares Modell" und stellt eine Programmiersprache bzw. ein Programmpaket dar. Das Programm selbst wurde in C geschrieben und liegt laut Handbuch auch in einer Fortran- und einer PL/I-Version vor.

Fuer welches Programm wollen Sie eine Maske haben ?

0 Direktdatei anlegen	3 Grenzueberschreitg	4 Haeufigkeitsvertlg
5 Beigeordnete Var.	5a NUB-Datei anlegen	6 NUB-Datei ausgeben
7 Sortieren	10 2-,3-dim. Tabellen	11 Mehrfach-Tabellen
13 Matrix-Operationen	14 Rasch-Skalierung	15 Lat.Strukt.analyse
18 Zellen-Mittelwerte	19 Korrelationsmatrix	20 Allg. lin. Modell
25 Pfadanalyse	28 Quad. Optimierung	30 Faktorenanalyse
32 Faktorwerte	33 Ähnlichk.skalierg	34 MDS (Kruskal)
36 Clusteranalyse		Info / Abbruch

ALMO ist jedoch keine Programmiersprache im Sinne von Modula-2 etc., sondern ein Statistiksystem. Es umfaßt das gut 557 kB(!) große Hauptprogramm sowie verschiedene weitere Dateien. Die verwandten Schlüsselwörter sind in deutscher Sprache gehalten. Das Programmpaket besteht aus einer doppelseitigen Diskette (formatiert mit 10 Sektoren pro Track und 82 Tracks pro Seite, also mehr als sonst üblich), von der auf normalem Wege (sprich mit den Desktopfunktionen) kein Backup angelegt werden kann. Dazu kommt ein gut 500 seitiges Handbuch in unhandlicher Spiralbindung, welches mit einem einfachen Textverarbeitungsprogramm (unproportionale Schrift) angefertigt wurde. Ferner war der vorliegenden Version noch eine 8 seitige Kurzanleitung mit grundlegenden Bedienungshinweisen beigelegt.

Aus dieser Kurzanleitung stammt der folgende, bemerkenswerte Abschnitt: "Sie können dann, ohne ALMO wieder verlassen zu müssen, eine Analyse nach der anderen rechnen. Allerdings geht das nicht beliebig oft, da der Speicher nicht zu 100% freigegeben wird. Es kann dann zu einem Absturz kommen."

Die Installation von ALMO auf einer Festplatte ist zwar nicht unmöglich, aber recht umständlich. Der erste Versuch "Ordner ALMO anlegen, Disketteninhalt hineinkopieren" scheitert an den festen Zugriffspfaden von ALMO, die im Gegensatz zu den (ansonsten üblichen) relativen Pfaden benutzt wurden. Für den Festplattenbetrieb wird empfohlen, trotzdem eine RAM-Disk anzulegen. Hierbei muß ein Buchstabe hinter denen der Partitions gewählt werden, etwa "H". Da der ST von bis zu 16 Laufwerken ausgehen kann, sollte man meinen, daß auch "P" gewählt werden könnte, aber dem ist nicht so, da im Programm von nur 8 statt der 16 Laufwerke ausgegangen wird.

Benutzt man keine Festplatte, sondern nur einen normalen ST mit einem Laufwerk (1 MB ist schon wegen der Programmgröße erforderlich; RAM-TOS ist nicht möglich, was aber nicht stört, da die meisten ST-Benutzer wohl doch das TOS im ROM haben), so kann mit einem mitgelieferten Programm eine 156kB große RAM-Disk eingerichtet werden. Platz für Accessories bleibt dann natürlich kaum noch.

Nach dem Booten mit einer Kopie der ALMO-Diskette steht die RAM-Disk zur Verfügung. Nun werden die Resource-Files, die bei Bedarf nachgeladen werden, und die Testdaten in die RAM-Disk kopiert und ALMO gestartet. Wer jedoch eine schöne GEM-Oberfläche erwartet (wie es in diversen ALMO-Verkaufsanzeigen suggeriert wird), dürfte sehr enttäuscht sein. Nachdem der Bildschirm gelöscht wurde, erscheint ein schlichtes Menü und sonst keine GEM-Elemente. Das gesamtes Auswahlmenü enthält gerade mal 9 Einträge incl. Hilfe, Programm-Info und Quit. Hier hätte man doch besser mal einen Blick auf diverse GEM-Programme und deren Gestaltung werfen sollen. Im Zuge der 19 "-Monito-

re und der vielen Farben, die der ST jetzt schon bietet und der TT demnächst bieten wird, ist es unabdingbar, daß Programme bei veränderten Hardwarekonfigurationen nicht mit Überraschungen aufwarten. ALMO läuft - wie in Test mit Protos zeigte - nicht mit Ganzseitenmonitoren zusammen. Außerdem wird wunderlicherweise mal mit deutschen Umlauten "ä", "ö" und "ü" gearbeitet, dann aber auch immer wieder mit "ae", "oe" und "ue" wie zu Computer-Zeiten, als es die passenden Buchstaben noch nicht gab.

Betrachten wir die einzelnen Menüpunkte. Klickt man die Programminformation an, so erscheint eine Meldung, wer welches Teilprogramm geschrieben hat. Immerhin werden hier fünf Autoren genannt. Diese Informationsdialogbox kann jedoch nicht in gewohnter Weise mit RETURN verlassen werden, man muß schon die Maus in die Hand nehmen.

Im nächsten Menü kann ein Inhaltsverzeichnis der Laufwerke A bis H (es sind ja nur 8 möglich!) abrufen. Nach Wahl der Station erscheint ein File-Selector, der zur Betrachtung des Inhaltsverzeichnisses dient. Analog wird auch bei den Menüpunkten "Löschen" und "Umbenennen" verfahren.

Das dritte Menü liefert einen Zugang zu Almo, den Masken und dem Editor. Das vierte Menü enthält nur einen Punkt: Hilfe.

Das vielversprechende dritte Drop-Down-Menü soll näher betrachtet werden. Dazu gehen wir streng nach der Reihenfolge der Kurzanleitung vor. Zuerst wird "Masken" angeklickt. (Mit "Maske" ist eine Auswahldialogbox gemeint. Der Begriff "Maske" wird eher auf Großrechnern benutzt. ALMO stammt laut Werbung vom Großrechner und wurde auf den ST portiert.) Daraufhin erscheint eine Dialogbox, wieder einmal ohne Default-Button (via RETURN anwählbar). Nach Wahl von "4 Häufigkeitsverteilung" wird PROG04.ALM erzeugt, welches auch sogleich gestartet werden kann. Die Ergebnisse können dann im Editor betrachtet werden. Der Editor hat einen ausreichenden Funktionsumfang, etwas ungewöhnlich ist wieder die Verwendung von GEM- und Nicht-GEM-Elementen und die Vorgabe einer festen maximalen Zeilenzahl für ein ALMO-Programm. Das Menü hätte an Übersichtlichkeit gewonnen, wenn ein paar Trennstriche mehr vorhanden wären und die Tastenkürzel am rechten Rande der jeweiligen Einträge stünden, wie es für GEM-Programme empfohlen wird. Ferner könnten unsinnige Einträge wie "+ = Shifttaste" entfallen. Diese Einträge sind nämlich anwählbar, jedoch wird durch Auswahl der Funktion offensichtlich keine Aktion ausgelöst.

Der Editor verfügt über Blockoperationen, die normalen Speicher- und Ladeoperationen sowie die Funktionen Suchen und Ersetzen. Etwas ungewohnt sind die Blätterfunktionen, die es ermöglichen, zu 10%, 20%, ... der Textlänge zu springen. Dies ist wohl als Ersatz für die fehlenden Marken gedacht. Ein anderer Text-Editor wie TEMPUS läßt sich nicht einbinden.

Das Programm macht rundum einen unfertigen Eindruck, besticht aber über die diversen enthalten Programme. Besonders störend im Umgang mit ALMO ist die Vermischung von GEM- mit herkömmlichen Programmelementen. Es wäre besser, man hätte konsequent ein GEM-Programm geschrieben. Ferner ist die Programmlänge erstaunlich groß.

```
#--------- Programm 4 ; Eindimensionale Haeufigkeitsverteilung ---------#
VEREINBARE
  Variable= 20  ;                          # Speicher fuer x Variable #

ANFANG                                                          Info1
Programm=4;
  Lese Namen_und_Beigeordnete_V  aus Datei 1  'D:\Testdat.nub'  ;

  Variable= V1:8                    ; # zu analysierende Variable#
  Option 1= V5:8                    ; # Mittwert+Stdabwg V5:8#
                                      # freier Raum fuer Optionen#
Ende_Programmparameter

Lese alle_V aus Datei 2  'D:\Testdat.dir'  Format direkt leer_zu ENDE;

                                                                Info2

Gehe_in_Programm                  # Raum fuer Umkodierungen #
Gehe_zu LESE
ENDE              Speichern unter  D:\PROG04.ALM                Abbruch
```

Man wird den Eindruck nicht los, daß hier platzverschwendend programmiert wurde. Soweit das Programm.

Die Programmiersprache von ALMO ist etwas anders geartet als die bekannten Programmiersprachen wie C oder Pascal. Zum einen fallen die deutschen Schlüsselwerte wie "LESE", "VEREINBARE", ... auf, zum anderen der für Programmierer ungewohnte Aufbau. Aber ALMO soll ja auch einen anderen Zweck verfolgen als die Programmierung mit C etc. Wenn man sich erst einmal an die Programmiersprache ALMO gewöhnt hat, dürften kaum Schwierigkeiten auftauchen

Das Handbuch ist mit gut 500 Seiten bei dem Programmumfang sicher nicht zu klein, jedoch wären ein Ringordner für die Handlichkeit und ein besserer Druck für die äußere Erscheinung wichtig. Auf etwa 20 Seiten wird eine Einführung und ein Überblick gegeben. Auf weiteren 170 Seiten wird die Programmiersprache ALMOs erklärt. Diese Einführung könnte jedoch noch wesentlich ausführlicher sein, denn gerade ein Anfänger wird damit nicht klarkommen. Die letzten fast 300 Seiten beschäftigen sich mit den in ALMO enthaltenen Programmen. Hier werden sogar Programme erklärt, die nicht einmal im Lieferumfang enthalten sind. Das Stichwortverzeichnis ist mit 4 Seiten etwas knapp.

Die Spanne der enthaltenen Programme reicht von einfachen wie der Überprüfung der Unter- und Obergrenzen bis hin zu komplexen Programmen wie dem allgemeinen linearen Modell. Für viele Zwecke ist ein Programm dabei, weshalb ich sie hier nur kurz aufzählen möchte: eindimensionale Häufigkeitsverteilung, Mittelwerte (nur arithmetisches Mittel), Standardabweichungen, Sortieren, zwei- und dreidimensionale Tabellenanalyse, beliebig-dimensionale Tabellierung, Matrix lesen/ schreiben/ ändern, Rasch-Skalierung und probabilistisches Unfolding, Analyse latenter Strukturen, Zellenmittelwerte und -streuungen, Korrelations-/ Kovarianz-/ Quadratsummenmatrix, Pfadanalyse für quantitative, ordinale und nominal-dichotomische Variablen, quadratische Optimierung, Faktorenanalyse, Faktor- und Gesamtpunktwerte, Ähnlichkeitsskalierung, nichtmetrische multidimensionale Skalierung, Clusteranalyse und natürlich die weiter oben genannten Programme zur Überprüfung der Grenzen und zum allgemeinen linearen Modell.

Hervorheben möchte ich das Programm "Allgemeines Lineares Modell". Dieses

mitgelieferte Programm ist sehr umfangreich und enthält 12 weitere Submodelle wie Regressions- oder Varianzanalyse. Die Erklärungen dazu sind ausführlich, was aber auch zu erwarten ist, denn gerade dieses Modell ist das Hauptanliegen von ALMO.

Bemerkenswert ist noch, daß sehr viele Fremdwörter benutzt werden, die jedoch die Fachfrau/ den Fachmann nicht stören sollten. Wo wir gerade dabei sind: Fachleute. ALMO ist zwar ein Statistiksystem, jedoch muß man sich erstens in der Statistik etwas auskennen und kann hier nicht so einfach drauflosarbeiten, und zweitens ist Statistik noch lange nicht Statistik.

Die mathematische Statistik selbst kommt bei ALMO etwas zu kurz. Auch der Naturwissenschaftler wird bei ALMO nicht gerade das finden, was er braucht. (Dies soll jedoch nicht heißen, daß man das Programm nicht auch als Naturwissenschaftler gebrauchen könnte.) ALMOs Zielgruppe sind eher Sozial- und

```
 Desk  Datei  Block  Suchen  Blättern  Hilfe  Drucke

#--------- Programm 4 : Eindimensionale Haeufigkeitsverteilung ---------#
VEREINBARE
  Variable= 20 ;                          # Speicher fuer x Variable #

ANFANG

Programm=4;
  Lese Namen_und_Beigeordnete_V aus Datei 1  'D:\Testdat.nub'  ;
  Variable= V1:8                          ; # zu analysierende Variable#
  Option 1= V5:8                          ; # Mittwert+Stdabwg V5:8#
                                            # freier Raum fuer Optionen#
Ende_Programmparameter

Lese alle_V aus Datei 2  'D:\Testdat.dir'   Format direkt leer_zu ENDE;


Gehe_in_Programm # Raum fuer Umkodierungen #
Gehe_zu LESE
ENDE
```

Wirtschaftswissenschaftler, wie die ALMO-Beispiele zeigen.

Summa summarum läßt sich sagen, daß ALMO ein recht umfangreiches System ist, jedoch diverse Unzulänglichkeiten in der Durchsetzung des Programms aufweist. Die Vermischung von GEM- und Nicht-GEM-Elementen ist gerade für die ST-Freunde sehr störend, aber wer darauf sowieso keinen Wert legt, der erhält mit ALMO ein umfangreiches Programm.

Dietmar Rabich



Wow! Soundliebhaber sollten bei **To be on Top** von "Rainbow Arts" hereinschauen. Computersoundvirtuose Jochen Hippel hat sich selbst übertroffen. Tolle Melodien und Effekte! Ziel des Soundspiels ist es, als armer Musiker einen Hit in den Top Ten zu landen. Es müssen Inspirationen gesammelt und ein Musikstück komponiert werden, das man dann im Fernsehen vorführen darf - vorausgesetzt, der Titel landet in den Top Ten auf den vorderen Plätzen. Selbst wenn man keine Noten lesen kann und auch sonst nicht sonderlich musisch begabt ist, lassen sich schöne Sounds erstellen. Vorsicht! Einige ST-Verkaufsversionen von To be on Top stürzen gleich im ersten Haus ab. Vorm Kauf einmal Probespielen.





To be on Top

Schöne Frauen, tolle Autos - Rolex am Arm. In die Welt der Yuppies führt das Strategiespiel **Yuppi's Revenge**. Programmiert wurde das Game vom Hanse-Autor Ralf Glau. Für Strategiespielverhältnisse wartet das Game mit einer hübschen Grafik auf.

In **Dragonscape** flattern Sie mit einem Drachen durch Fantasielandschaften, sammeln Gegenstände ein, die dann zum richtigen Ort gebracht werden müssen. Grafik: Hm! Sound: Na ja! Spielspaß: Wo?





Dragonscape

Roadblasters ist ein Autorennen, bei dem es hart zur Sache geht. Konkurrenten werden einfach mit der Kanone von der Straße gepustet. Grafik und Sound sind bei "U.S Golds" neuer Automatenumsetzung überdurchschnittlich!





Roadblasters

Bei **Roy of the Rovers** von "Gremlin" handelt es sich um eine Mischung aus Adventure und Fußballspiel. Im ersten Programmteil muß man als Mannschaftskapitän fünf entführte Spieler suchen. Im zweiten Teil wird dann gekickt. Für ein oder zwei Spieler! Schade, daß der Computerkicker so schwach spielt!

Sierra scheint Adventures am Fließband zu programmieren: Larry 2, Police Quest 2, Gold Rush, Manhunter und jetzt auch **Kings Quest 4**. Im Vergleich zu den anderen Kings Quest-Adventures gibt es detaillierte Grafiken und einen besseren Sound. Ansonsten bleibt alles beim alten (Parser, Spielablauf!) Nur die Handlung ist neu! Wie bei allen Sierra-Adventures hat der Spieler viele Stunden Spielspaß vor dem Monitor zu verbringen.



Roy of the Rovers

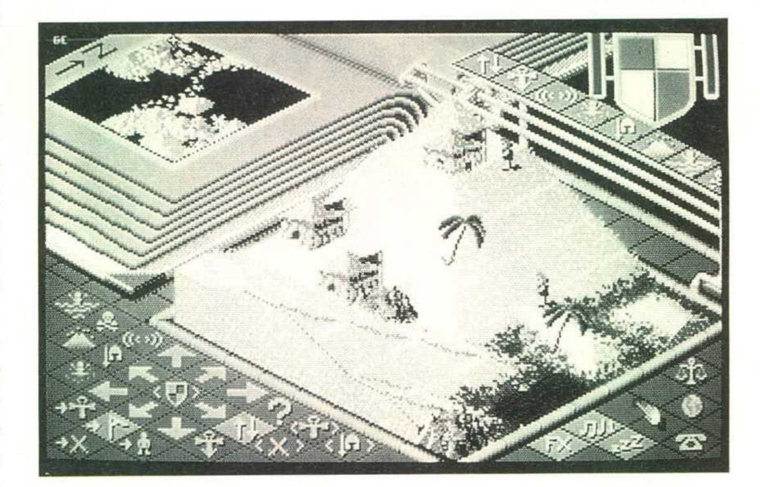


Kings Quest 4

Kick off heißt das erste gute Produkt aus dem Hause "Anco": ein tolles Fußballspiel im Stil von "Microprose Soccer" auf dem C64. Schnelles Scrolling, passable Grafik und hitziges Gameplay. Kick off ist das derzeit beste Fußballspiel für den ST. Zugreifen lohnt sich!

Populous

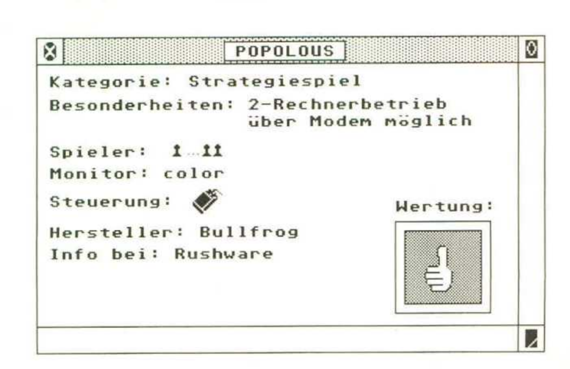
Mit Populous hat Bullfrog eine neue Dimension von Computerspielen eröffnet. Logisch, daß Bullfrogs Chefprogrammierer Peter Molyneux in diesem neuen Universum keine Giganten, sondern Gottheiten miteinander konkurrieren läßt.
Populous ist dabei eine gelungene Mischung aus Strategie-, Action- und Glücksspiel. Der Gradmesser des Erfolges ist - wie sollte es bei Göttern anders sein - der Glaube der Bevölkerung an den "richtigen Gott".Die Bevölkerung wird durch den Spieler indirekt gelenkt. Die Hilfsmittel sind dabei die in diesem Metier üblichen. Der Schöpfer richtet sich seine eigene Welt ein, indem er Berge, Täler und Flüsse entstehen, Wüsten zu Weiden werden läßt und dem Meer neues Land abnimmt. Damit sich die Bevölkerung fleißig mehre, kann er je nach Entwicklung der Population von der Hütte bis hin zum Schloß für die entsprechenden Unterkünfte sorgen. Je schneller die Häuser überbelegt sind, desto eher werden sich die Bewohner nach neuem Lebensraum umschauen und neues Territorium für sich und damit für ihren Gott erschließen.



Auf der anderen Seite der Beeinflussungsskala stehen die noch im Mittelalter anerkannten Gotteszeichen: Martialische Krieger, Naturkatastrophen wie Dürre oder Vulkanausbrüche führen zu unangenehmen Folgen für die Population. Repressionen sorgen bekanntlich dort, wo Gutes nicht mehr wirkt, oft noch für den angestrebten Erfolg...

Allein die große Variationsvielfalt - immerhin besteht die "Welt" aus 10.000 Szenarien-, die detailliert ausgearbeiteten Graphiken und der plastische Sound sind begeisternd und zeugen von der professionellen, sechsmonatigen Arbeit des Teams von Bullfrog.
Die neue Dimension von Populous beginnt auch jenseits dieser Bewertung. Die aufwendige perspektivische Graphik erfordert für den Betrieb mit zwei Spielern zwei Rechner. Diese werden einfach über die serielle Schnittstelle verbunden. Zur weiteren Steigerung will Bullfrog im Spätjahr den Erweiterungssatz Populous Plus mit zusätzlichen Szenarien und Computerstrategien herausbringen. Ebenfalls in Arbeit ist eine Ausgabe, die es bis zu 16 Spielern im Netzwerkverbund erlaubt, gegeneinander zu spielen.

CPL


POPULOUS
Kategorie: Strategiespiel
Besonderheiten: 2-Rechnerbetrieb über Modem möglich
Spieler: 1..11
Monitor: color
Steuerung:
Hersteller: Bullfrog
Info bei: Rushware
Wertung:



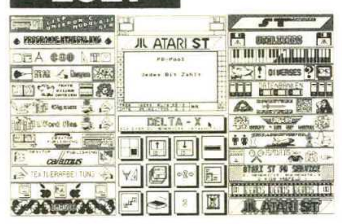
23 PD-Anbieter aus dem gesamten Bundesgebiet haben wieder viele gute Programme in den PD-Pool gegeben und daraus die neuen Disketten der gemeinsamen Serie zusammengestellt. Die vorangegangenen Disketten - 2001 bis 2020 - sind ebenfalls vorrätig und sofort lieferbar, Programmbeschreibungen dazu finden Sie in Heft 5/89 S. 24 + 25, bzw. Heft 6/89 S.182 + 183. Der Anbieter in Ihrer Nähe informiert Sie gerne über weitere Programme in der Public Domain, Anruf genügt.

PD- und Shareware-Autoren können neue und leistungsfähige Programme im Rahmen der PD-Pool Serie veröffentlichen. Bitte sprechen Sie mit einem der nebenstehenden Anbieter, wenn Sie Ihrem Programm von Anfang an zum Erfolg verhelfen wollen.

Viel Spaß mit den neuen PD's

Die "23"

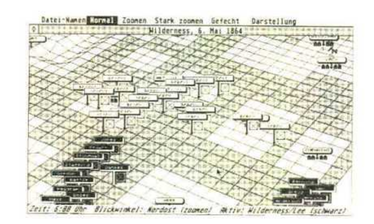
2021

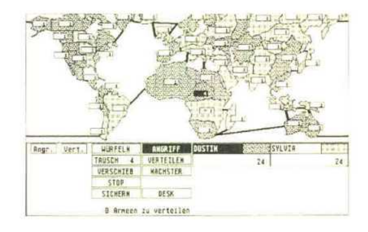


Delta X setzt neue Masstäbe beim Bedrucken von Zweckform Disklabels. Übersichtliche Bedienung, große Grafikbibliothek, STAD-, Degaskompatibel, druckt auf Epson-, bzw. NEC P6-kompatiblen, toll! (s/w)
Barcode druckt EAN 8 und 13.
Fonts für Proportionalschrift unter Starwriter. Esperantozei., umfangreiche Dokumentation und Installationshilfen.
VIZZ-2, Video-Zeit/Zählwerk-Umrechnung ermittelt Band- und Restlaufzeiten. (s/w)
Disk 5.98: Diskverwaltung mit umfangreichen Sortier und Druckmöglichkeiten. (s/w).
Database:einfache, schnelle Datenbank mit Serienbrief, Import, Export und vielen nützlichen Funktionen.
Druck 31 bringt Texte, Listings und Bilder sauber zu Papier. Schriftarten, Zeichenbreiten, Zeilenabstände, Druckdichten etc... sind verstellbar.
Epson NL Hardcopy-Ausdruck mit 960Zeichen.

2022

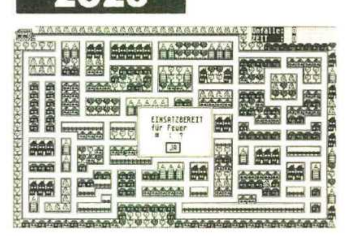
Empire: 3 neue Scenarios
Wilderness: UMS Scenario.



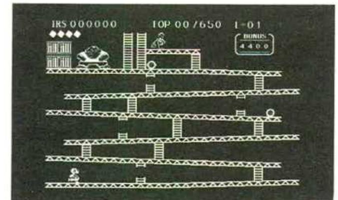


Wagnis: Strategiespiel für 2-6 Spieler, bei dem Sie die Welt von gegnerischen Armee befreien müssen. (s/w)
Eamon Adventure Set, bestehend aus Basismodul und 5 Abenteuern. Sämtliche Source Codes (GFA) werden mitgeliefert, damit Sie eigene Programme nach ähnlichem Muster schreiben können.
Daleks: Robotvariante. (s/w)

2023

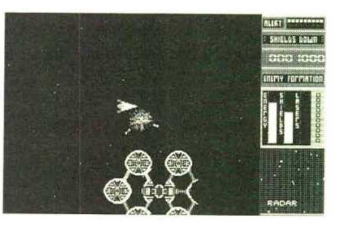


Feuerwehr fordert Ihr ganzes Können. Lenken Sie das Einsatzfahrzeug sicher zum Brandherd und zurück. (s/w)
Niemals: oder doch? Spielhit aus dem gleichnamigen Film, Niemals verlangt blitzschnelle Reaktionen. (s/w)
Monkey: ein Klassiker! (s/w)



2024

Starbattle: Weltraumballerei.
Azarian: Erinnerungen an Megaroids werden wach und dann geht's richtig los! (f)

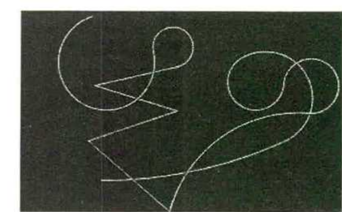


Spacewar: Strategie & Action für 2 Weltraumpiloten. (f)
1000 Meilen Autorennen. Strategiespiel mit komplettem Source Code in GFA-Basic. (f)
Stoneage: Arcadespiel mit eigenem Leveleditor und viel Spielspaß. (f)
PAC-Manic jetzt wieder in. (f)

2025



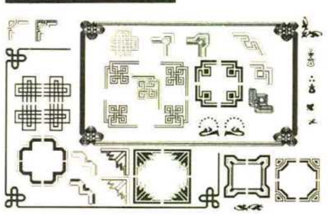
Grafikserver erledigt nützliche Arbeiten im Grafikbereich.
Fineline zeichnet schöne glatte Kurven! Degas/NEO-Kompatibel, toll! (f)



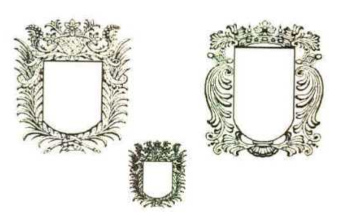
C-Chrome Malprogramm: vielseitig, leicht erlernbar, kompatibel und schnell! (f)
Sigipainter akzeptiert sämtliche Fremdformate und malt in allen Auflösungen.
Amok besticht durch einige verrückte Funktionen und Effekte. (s/w)

(f) = Farbe
(s/w) = monochrom

2026



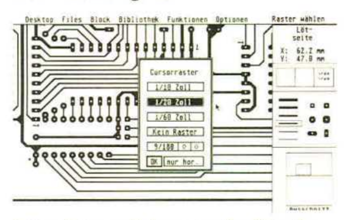
Clip-Art-Album: wir haben fünf normale PD-Disketten auf eine gepackt und dazu ein Menü geschrieben, mit dem Sie die vielen Grafiken übersichtlich verwalten können. (s/w)



2027



A-Generator erstellt Alert-Boxen für Basic & C. (s/w)
GFA-Edit 1.8 die Neue!
PCB Edit Platinenlayout mit 180 DPI Auflösung, gängigen Rastergrößen und einfacher Handhabung. (s/w)

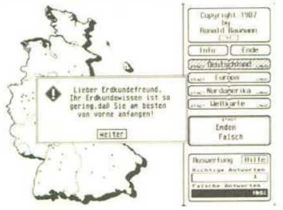


Strukto zeichnet selbständig Struktogramme von GFA-Basic(2.0)-Programmen. (s/w)

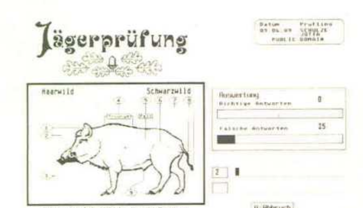
Trapper überwacht GEMDOS, BIOS und XBIOS-Aufrufe.
3D-Editor ist eine von 5 Demos in GFA 3.0 (s/w).

2028

Bruchrechnen f. Schüler. (s/w)
Mathquiz druckt Übungsaufgaben und separate Lösungsbögen der vier Grundrechenarten aus. Die Größe der verw. Zahlen ist frei wählbar.
Erdkunde fragt Bundesländer und Städtenamen ab. (s/w)



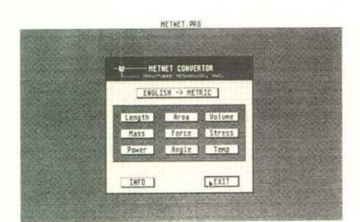
Jagd bereitet Sie auf die Jägerprüfung vor. (s/w)
Ratlefax: Tiernamen-Ratespiel m. 3 Schwierigkeitsstufen. (s/w)
W.D.E. Multiple Choice



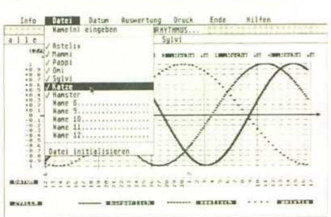
Abfrageprogramm mit 422 Fragen aus Geographie, Kunst, Geschichte, Naturw., Sport und Allgemeines. Fragenkatalog erweiterbar. (s/w)

2029

Pegafakt: lauffähig Demo des Fakturierprogramms. (s/w)
Metnet: Umrechnungshilfe



English-Metric, für Maße, Temperaturen, Gewichte und mehr. Läuft auch als Accessory! (s/w)
General Ledger: Kassenbuch, mandantenfähig, Passwort, Menüsteuerung, versch. Auswertmöglichkeiten. (s/w)
Wirtschaftsrechnen für Bürokaufleute. (s/w)
PH-Wert Berechnung für Säuren/Basen, Ampholyte, Salze und Puffersysteme. (s/w)
Bio 2.6 Rythmusanalyse der Spitzenklasse. Sauber gestaltet, leicht bedienbar, gute Grafik und Druckmöglichkeiten. (s/w)



El-Analyse errechnet empirische bzw. Summenformeln, Molmassen. (s/w)

2030

VIP Datenbank mit 1000 Namen & Geburtstagen berühmter Persönlichkeiten.
Autobörse Gebrauchtwagen-Vermittlungsprogramm, mit dem Sie einen Nebenerwerb aufbauen können. Das komplett ausgearbeitete Geschäftskonzept ist beim Autor erhältlich.
STandard Base III: lauffähige Demoversion, mit der die Datenbanken VIP und Autobörse betrieben werden.

Alle Programme garantiert virenfrei — **G DATA** Anti Viren Kit II

Besuchen Sie uns auf der ATARI-Messe in Düsseldorf, 25.-27.08.89

Die vorgestellten Disketten erhalten Sie exklusiv bei folgenden PD-Anbietern:

Die Schnittstelle Soldiner Straße 14 1000 Berlin 65 Tel: 030/4948820	**H&S Wohlfahrtstätter** Irenenstr. 76c 4000 Düsseldorf 30 Tel: 0211-429876	**Digital Image GbR** Postfach 1206 D-6096 Raunheim a.M. 06142/22636 od. 43560	**Weeske Computer** Potsdamer Ring 10 7150 Backnang 07191/1528-29 od. 60076	**Software-Service Duffner** Ritterstr. 6 7833 Endingen a. K. Tel: 07642/3875
V.U. - Volker Uecker Hohenkamp 2 2308 Preetz Tel: 04342-83842	**OHST-Software** Nelkenstr. 2 4053 Jüchen 2 Tel: 02164/7898	**IDL Software** Lagerstraße 11 6100 Darmstadt 13 06151-58912	**Gauger Software** Buhlstr. 16a 7505 Ettlingen Tel: 07243 - 31828	**Karolin Lauterbach** Josephsplatz 3 8000 München 40 089-2722377
ST Profi-Partner Mönkhofer Weg 126 2400 Lübeck Tel: 0451/505367	**Elektronikversand Michiels** Leloh 24 4056 Schwalmtal Tel: 02163/4187	**Computer Treff** Nettelbeckstr. 12 6200 Wiesbaden Tel: 06121 - 404302	**iks** Schönblickstr. 7 7516 Karlsbad 4 ab 18.00 Uhr 07202/6793	**Andreas Meyer PD** Frauenhofenerstr. 29 8255 Schwindegg 08082-5125 (ab 15 Uhr)
T.U.M.-ST-Soft Postfach 1105 2905 Edewecht Tel: 04405/6809	**Intersoft** Nohlstr. 76 4200 Oberhausen 1 0208-809014	**Computer-Software Markert** Balbachtalstr. 71 6970 Lauda 9 Tel: 09343/3854	**=PD-EXPRESS= J. Rangnow** Ittlinger Straße 45 7519 Eppingen-Richen 07262/5131 (ab 17 Uhr)	**Graf & Schick EDV** Hauptstraße 32a 8542 Roth Tel: 09171/5058-59

☐ Scheck über DM ______ liegt bei, ich erhalte die Ware verpackungs- und versandkostenfrei.

☐ Per Nachnahme. Nur Inland! (zuzüglich DM 6,- Nachnahmegebühr).

☐ 2001	☐ 2011	☐ 2021	Wenn Sie weitere Disketten aus unseren Sammlungen - bzw. aus der P.D. Fibel - bestellen möchten, geben Sie bitte die gewünschten Diskettennummern an:
☐ 2002	☐ 2012	☐ 2022	
☐ 2003	☐ 2013	☐ 2023	
☐ 2004	☐ 2014	☐ 2024	
☐ 2005	☐ 2015	☐ 2025	☐ ______
☐ 2006	☐ 2016	☐ 2026	☐ ______
☐ 2007	☐ 2017	☐ 2027	☐ ______
☐ 2008	☐ 2018	☐ 2028	☐ ______
☐ 2009	☐ 2019	☐ 2029	☐ ______
☐ 2010	☐ 2020	☐ 2030	☐ ______
Beschreibung: Siehe Heft 5/89 Seite 24 & 25	Beschreibung: Siehe Heft 6/89 Seite 182 & 183		

Diskpreis: 8,- DM

unverbindlich empfohlener Verkaufspreis

Lieferanschrift:

____

____

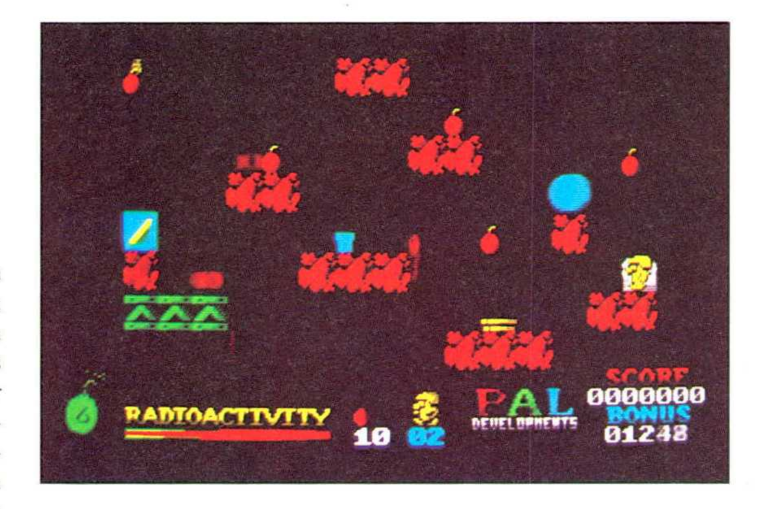
____

____

Gewünschte Disketten ankreuzen und Bestellschein an einen der oben angegebenen Anbieter einsenden.

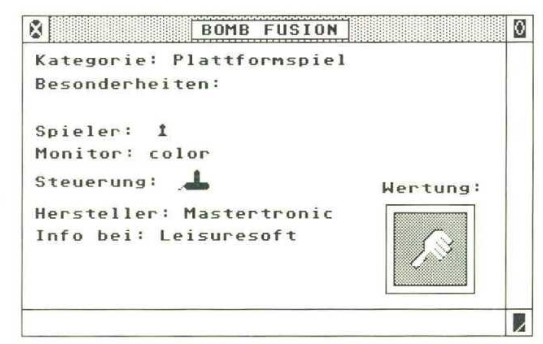
Bomb Fusion

Ein Profi muß her! Terroristen haben Bomben in einem Atomkraftwerk gelegt. Obendrein versagt das Computersystem, das für die Wiederaufbereitung der verbrauchten Brennstäbe zuständig ist. Aufgabe des "Bomb Fusion"-Spielers ist es, den atomaren Treibstoff einzusammeln und ihn dann in Transportkisten zu verfrachten. Brennstäbe werden aus einem Rohr ausgestoßen und irren ziellos umher. Der Spieler sammelt sie durch Berührung ein. Wenn die Zündschnur einer Bombe brennt, heißt es, schnell dorthin zu springen. Durch eine einfache Berührung wird die Bombe entschärft. Gelingt es Ihnen nicht, rechtzeitig eine aktive Bombe zu erreichen, macht es "BUMM". Allerdings nur grafisch. Keine Panik, Ihr Computer bleibt heil. Der Bildschirm blitzt nur einmal kurz auf. Drei Explosionen kosten ein Bildschirmleben. Am unteren



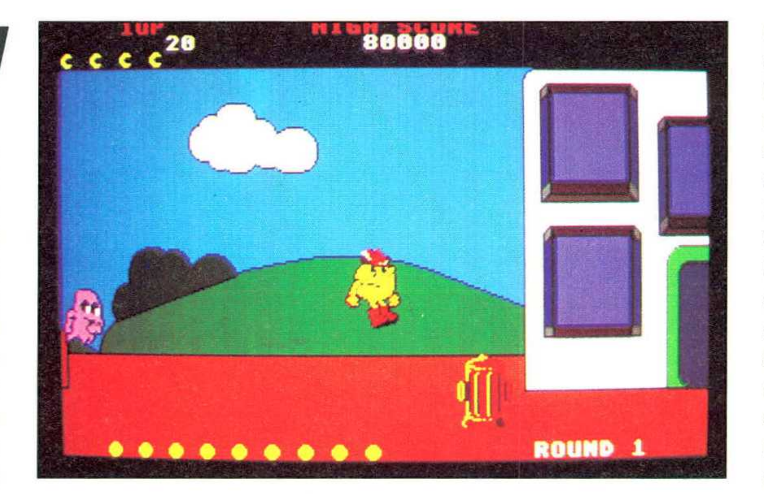
Bildrand befindet sich ein Bombensymbol, das die Zeit bis zum Knall anzeigt. Daneben ist ein Balken angebracht, an dem sich ablesen läßt, wie stark der Held radioaktiv verseucht ist. Steht diese Anzeige auf Maximum, sind Sie ebenfalls ein Leben los. Doch es gibt noch andere Gemeinheiten: Berührt der Held eine große Kugel, die sich über den Bildschirm bewegt, steigt die radioaktive Verseuchung. Außerdem nimmt sie ihm die eingefangenen Brennstäbe wieder ab. Gegen die Kugel ist kein Kraut gewachsen. Man kann ihr höchstens ausweichen. Mit zehn Brennelementen beginnt das Spiel. Pro Level steigert sich diese Zahl um zwei. Bei Null wird es Zeit, das Level zu verlassen. Das geht nämlich nicht automatisch. Der Profi muß selbst zum Ausgang. Dieser ist als Gitter dargestellt. Waren Sie schnell genug, gibt es noch einen Punktebonus. Der Profi kann nicht alle Stellen des Screens erreichen. Hier helfen ihm Trampolins: Springt er auf eines, kann er auch höher gelegene Plattformen erreichen. Fällt der Spieler unten aus dem Bildschirm, kommt er oben wieder herein. So gelangt man zwar auf alle Plattformen, jedoch kostet diese Methode sehr viel Zeit. "Bomb Fusion" ist grafisch schlicht aufgemacht. Es gibt nur wenige verschiedene Plattformtypen. Eine schrille Melodie begleitet das Spiel. Sie ist glücklicherweise auch abschaltbar. "Bomb Fusion" ist zwar kein Renner, aber immer noch besser als Mastertronics andere Neuerscheinung "Chase". Ein schwacher Trost!

Carsten Borgmeier



PacLand

Pac Man ist wieder da! Diesmal mit Armen und Beinen! Richtig fein hat er sich gemacht, mit Mütze, Handschuhen und Stiefeln. Er trägt eine kleine Fee unter dem Hut und zieht mit ihr durch die Lande, von links nach rechts. Dabei wird er von seinen Erbfeinden verfolgt. Die Geister haben dazugelernt. Sie kommen jetzt in Flugzeugen angebraust und lassen Mini- Geister auf den Körnerfresser fallen. Oder sie kommen in Autos daher und versuchen, Pac Man zu überfahren. Für solche Fälle gibt es die guten alten Kraftpillen. Die Pillen stehen allerdings kurz vor dem Ende jeder Runde. Hat er eine Pille geschluckt, werden die Geister für einen Augenblick blau. Pac Man kann sie dann fressen. Die kleine gelbe Kreissäge (So sah er früher einmal aus, erinnern Sie sich?) wird dann noch von Hydranten, Baumstümpfen, Kakteen und an-



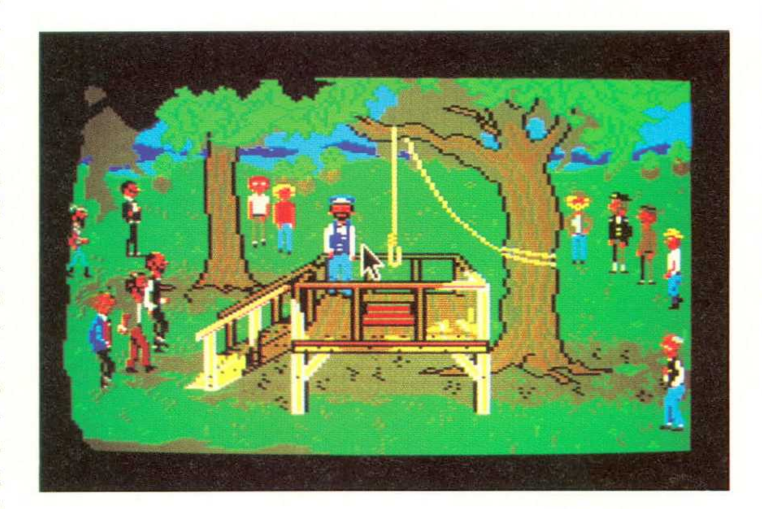
deren Kleinigkeiten aufgehalten. Mit dem Feuerknopf werden diese Hürden aber leicht übersprungen. Einige Gemeinheiten sind in die Landschaften eingebaut: Die Gebirge sind von Schluchten unterbrochen. Über diese Schluchten führen Holzstämme, die sich fahrstuhlartig auf- und abbewegen. Ein großer Swimming-Pool versperrt den Weg. Man kann ihn nur mit Hilfe eines Trampolins überspringen. Hier ist das Geschick gefordert, denn diese Stellen sind sehr schwer. Außerdem läuft Pac Man gegen die Zeit. Hat der Spieler eine Runde geschafft, erhält er die verbliebene Zeit als Punktebonus. Während einer Runde erscheinen Kirschen, die, eingesammelt, auch Punkte bringen. "Pac-Land" bringt frischen Wind in die schon recht eingefahrene Pac Man-Szene: Der Körnerfresser ist endlich aus seinem Labyrinth heraus. Dazu gibt es zeichentrickartige Grafik mit schnellem, horizontalem Scrolling (das aber manchmal ruckelt) und lustige Musik. Besonders Kinder (auch große!) werden daran ihre Freude haben. Bis auf einige wirklich gemeine Stellen ist dieses Game einfach toll zu spielen. Nicht nur für Pac Man Enthusiasten.

Carsten Borgmeier



Im Goldrausch

Haben Sie sich nicht immer schon einmal gewünscht, Gold zu finden? Das können Sie jetzt: Mitte des 19. Jahrhunderts bricht in Kalifornien ein Goldrausch aus. Viele zieht es zum Reichtum durch das glänzende Metall. So auch den jungen Jerrod. Er lebt in Brooklyn, New York. Ein Brief von seinem Bruder gibt schließlich den Ausschlag, sich auf eine waghalsige Reise zu begeben. Jerrod hat gleich drei Möglichkeiten, in den wilden Westen zu gelangen: Er kann in einer Kutsche quer durch die Staaten fahren. Oder er fährt mit dem Schiff ums Kap Horn herum (den Panama-Kanal gab es noch nicht!). Als dritte Alternative führt sein Weg zum Reichtum per Schiff bis nach Panama. Von dort aus schlägt er sich die Küste entlang bis nach Kalifornien durch. Jeder Weg ist ein eigenes Adventure! Kein Wunder, daß fünf Disketten gefüllt sind! Wie bei Sierra-Adventures üblich, läßt sich der Programmcode auf Festplatte übertragen. Dieser Schritt ist sehr empfehlenswert! Die Programmierer haben anscheinend mit dem alten



Entwicklungssystem gearbeitet: "Gold Rush" lädt jede Szene nach. Auch grafisch ist es nicht auf der Höhe der Zeit: Grobe Bilder erscheinen auf dem Monitor. Tastatureingaben werden am unteren Bildrand gemacht. Dadurch wird der Bildschirm noch etwas verkleinert. Dafür gibt es die gute alte Stick-Steuerung zurück. Sound war noch nie eine Stärke von Sierra. So auch hier: einfache Melodien und Effekte. Diese passen aber sehr gut zum Thema. Was die technische Aufmachung angeht, ist "Gold Rush" ein Rückschritt gegenüber LARRY 2 oder King's Quest IV. Atmosphäre macht sich dafür um so mehr breit: Zum Programm gibt es eine große Karte mit eingetragenen Reiserouten. Ein Taschenbuch führt in die Historie des Goldrausches ein. Sierra hat mit "Gold Rush" ein sehr realistisches Adventuregame geschaffen. Nicht nur für Geschichtsbesessene!

Carsten Borgmeier



GOLD RUSH

Kategorie: Adventure
Besonderheiten: Basiert auf histor. Begebenheiten
Spieler: 1
Monitor: color, monochrom
Steuerung:
Hersteller: Sierra
Info bei: Ariolasoft
Wertung:

Chase



Böse Buben entführten eine Weltraumprinzessin namens Chardonnay. Jetzt muß ein Held her, der das holde Mädel befreit. Mit einem Raumschiff nehmen Sie die Verfolgungsjagd auf. Soweit zur Hintergrundgeschichte von "Chase", "Mastertronics" neuestem Billigspiel für den ATARI ST. Besagte Verfolgungsjagd besteht aus vier Phasen: Zuerst geht es durch einen Asteroidensturm. Jeder Zusammenprall sorgt für den Verlust eines Schutzschilds. Zehn Schilde haben Sie zu Beginn. Die Entführer fliegen in blauen Raumschiffen vor Ihnen her. Fünf Treffer brauchen Sie, um ein Schiff zu sprengen. Ein Radarschirm am unteren Bildrand erleichtert das Zielen. Sie können auch ein Visier in das Cockpitfenster einblenden. Die zweite Phase ist eine Art Slalomfahren. Es gilt, zwischen zwei Säulen hindurchzufahren. Zehn solcher Slalomtore stehen dazu bereit. Haben Sie die hinter sich gelassen, beginnt die dritte Phase: ein Tunnel. Er wird ähnlich wie beim alten C64-Klassiker Magic Carpet durch herannahende Quadrate dargestellt. Ein einziger Crash mit den Rändern sorgt dafür, daß es in Phase vier weitergeht (unlogisch!): In Star Wars-Manier fliegen Sie dort durch einen Korridor und ballern mit Hilfe des Fadenkreuzes feindliche Schiffe ab. Doch Vorsicht! Rote Rechtecke versperren den Weg. Riskante Ausweichmanöver sind angesagt. Ist auch diese Phase geschafft, geht das Spiel wieder von vorne los. Mit dem Unterschied, daß in einigen Stufen die Angreifer andere Formen besitzen. Gesteuert wird das Raumschiff per Maus. Mit der linken Maustaste feuert man, mit der rechten wird beschleunigt und gebremst. An einigen Stellen reagiert die Steuerung sehr schwerfällig (das gibt Abzüge in der B-Note!!!). Grafisch stellt "Chase" durchschnittliche Softwarekost dar: nicht ausgefüllte Vektorgrafik, einfache Formen und wenige Farben. Sound ist nur in Form von Geräuscheffekten vorhanden. Doch es gibt noch mehr zu bemängeln: Die deutsche Anleitung ist ein Witz, was den Informationsgehalt angeht. Sie beschreibt lediglich die Kontrollmöglichkeiten. Auch wenn "Chase" zum günstigen Preis von ca. 35,- DM angeboten wird, lohnt sich ein Kauf nicht. Selbst für ein Billigspiel ist "Chase" äußerst schwach.

Carsten Borgmeier

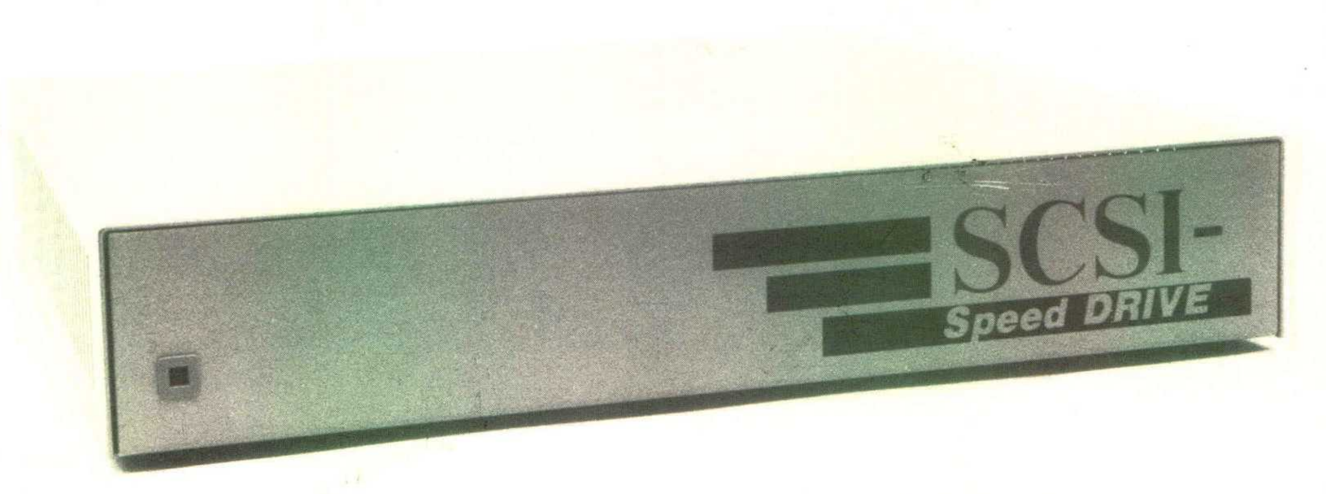


CHASE

Kategorie: Actionspiel
Besonderheiten:
Spieler: 1
Monitor: color
Steuerung:
Hersteller: Mastertronic
Info bei: Leisuresoft
Wertung:

SCSI Speed Drive Festplatten siehe Testbericht 68000 ST Magazin 5/89

... eine der schnellsten und leisesten Festplatten für den Atari ST ...



— 1 Jahr Garantie —

ALLE PREISE UNVERBINDLICHE PREISEMPFEHLUNGEN





Leistungsdaten

Die Verbindung eines reinen SCSI Hochgeschwindigkeits-Hostadapters und die Verwendung von SCSI Festplatten ermöglichen Geschwindigkeiten, die bisher noch nicht erreicht wurden.

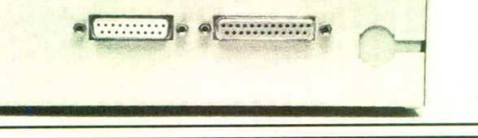
In der Praxis ergeben sich Geschwindigkeitssteigerungen zwischen 30 — 60 %.

Die Festplatte ist 100 % kompatibel zu den original Atari ST Festplatten. Das heißt: sie können auch andere Harddisktreiber oder den original Atari Harddisktreiber benutzen. Auch Programme wie PC Ditto, Aladin usw. sind auf unserer Festplatte selbstverständlich lauffähig.

Desweiteren ist in der Festplatte eine **Echtzeituhr** integriert.

Die Festplatte wird mit einer sehr umfangreichen Software ausgeliefert. Einige Leistungsdaten: Auto-Boot-fähig, Anlegen bis zu 12 Partionen, Bootpartion frei wählbar...

Zur weiteren Geschwindigkeitssteigerung wird das von uns entwickelte **Cache Programm „SPEED CACHE"** mitgeliefert.





DMA Port

Der DMA Port der Festplatte ist herausgeführt und komplett gepuffert. Das macht den Anschluß weiterer DMA Geräte (Atari Laserdrucker, weitere Festplatten etc.) möglich.

Die Festplatte stellt intern einen kompletten SCSI Bus zur Verfügung. Dieses ermöglicht den Anschluß weiterer SCSI Hardware.

Die Technik

Das Gehäuse der Festplatte wurde noch einmal überarbeitet. Durch eine besondere Art der Luftzirkulation wird die Festplatte ohne störenden Lüfter betrieben und die Laufgeräusche der Festplatte optimal unterdrückt. **Das macht diese Festplatte zu eine der leisesten Festplatten für den Atari ST.**

Das Gehäuse entspricht den Gehäuseabmessungen des Mega ST. Durch die robuste Ausführung kann es auch als Monitoruntersatz verwendet werden.

Das Netzteil (VDE, GS) verfügt über 65 W und kann auch eine zweite interne Festplatte versorgen.

Alle Festplatten verfügen über einen AUTO Park und sind mit einer **speziellen Pufferung** ausgestattet, die vor Schäden der Festplatte schützen, die durch kleine Stöße entstehen können.

Garantie, Service

Auf unsere Festplattensysteme gewähren wir **1 ganzes Jahr Garantie.**

Im Falle eines Defektes während der Garantiezeit wird die Festplatte von unserem Zustelldienst UPS (United Parcel Service) bei Ihnen abgeholt. Die Kosten für diesen zusätzlichen Service tragen selbstverständlich wir.

Qualität

Garantie

Service

Preise

32 MB 40 ms	50 MB 28 ms	85 MB 28 ms
1398.-	1798.-	2298.-

Hard & Soft A. Herberg, Bahnhofstr. 289, 4620 Castrop-Rauxel, Tel. 0 23 05/1 57 64, Fax. 12022 — Händleranfragen erwünscht —

The Real Ghostbusters

Die Geisterjäger sind bei uns durch den großen Kinoerfolg bekanntgeworden. Bei "The Real Ghostbusters" handelt es sich um eine Automatenumsetzung des lustigen Filmstoffs: Es gilt, Geister einzufangen und aufzubewahren. Das Handwerkszeug dazu schleppen Sie mit sich herum: ein Gewehr und einen Protonenstrahler. Beide Waffen werden per Feuerknopf ausgelöst: Mit einem kurzen Druck schießen Sie alle Gestalten ab. Daraufhin erscheinen die Geister, die mit einem langen Knopfdruck in den Rucksack gesaugt werden. Der Protonenstrahler muß wieder aufgetankt werden. Dazu gibt es Bonussymbole wie Ölfässer, Heizkessel oder auch Schubkarren. Einmal anschießen, drübermarschieren, und weiter geht die muntere Jagd. Auch ein paar Geister dienen als Lieferanten für die Extras: Neben einer Tankfüllung stehen noch höhere Schußkraft, eine unverwundbare Aura für begrenzte Zeit und ein freundlicher Geist auf der Liste der Extras. Der freundliche Geist kreist um den Spieler und bildet so einen Schutzschild. Nur die Aura verhindert wirkungsvoll einen Angriff aus allen Richtungen. Berührt der Spieler eine Gestalt, hat er ein Leben weniger. Einige Gestalten schießen sogar. In jedem Level hocken ein oder mehrere besonders fiese Typen, die den Ausgang bewachen. In einem Zwischenbild werden die gefangenen Geister in einer Apparatur eingesperrt, wo sie kein Unheil mehr anrichten können. Dann geht es im nächsten Level weiter. Man kann alleine oder zu zweit auf Geisterjagd gehen. Im Zwei-Spieler-Modus laufen beide Spieler gleichzeitig auf dem Bildschirm herum. Ähnlich wie bei "Gauntlet" sieht man die Action aus der Vogelperspektive. Leider ist "The Real Ghostbusters" ziemlich schwer. Ein Schuß der Gestalten nur knapp am Kopf des Helden vorbei - und schon ist ein Bildschirmleben futsch! Zehn abwechslungsreiche Level sind vorhanden. Eine Vielzahl an Monstern wie beispielsweise Werwölfe, elefantenähnliche Wesen, häßliche grüne Fratzen, Würmer oder das typische Bettlaken-Gespenst bevölkern die Spielfelder. Dagegen sehen die Spielerfiguren etwas kläglich aus: Gewehr im Anschlag mit grober Grafik. Genauso, wie sie aussehen, gehen die Helden auch: mit merkwürdig breiten und abgehackten Bewegungen. Dagegen ist die Musik ein Genuß: eine gute Umsetzung von Ray Parker Jr.s "Ghostbusters". Sieht man von dem übertrieben hohen Schwierigkeitsgrad und der teilweise schlampigen Grafik beim Helden ab, bleibt ein zufriedenstellendes Spiel übrig. Nicht mehr und nicht weniger!



Carsten Borgmeier


THE REAL GHOSTBUSTERS

Kategorie: Actionspiel
Besonderheiten:

Spieler: 1...2
Monitor: color
Steuerung:
Hersteller: Activision
Info bei: Ariolasoft

Wertung:



KLI-KLA-KLEISTER

Eines kann man über Floppies, Festplatten und deren weitläufige Verwandtschaft sicher sagen: Sie verursachen massenhaft Probleme. Zumindest, wenn man nach den Briefen der Leser des SCHEIBENKLEISTERs und dieser Zeitung urteilt. Häufige Fragen, die wir als Autoren des SCHEIBENKLEISTERs gestellt bekommen, sollen künftig regelmäßig hier veröffentlicht, beantwortet oder zur Diskussion gestellt werden. Wohlgemerkt: Hier geht es zwar "nur" um Massenspeicher, aber das ist immer noch ein weites Feld.

Gleich zu zwei bisher unbeantworteten Leserfragen:

- "Im SCHEIBENKLEISTER ist beschrieben, wie man in einem TEAC FD55F-Laufwerk zwischen 40 und 80 Spuren umschaltet, indem man einen Widerstand (R15) über einen Schalter anschließt. In meinem Laufwerk dieses Typs ist der betreffende Widerstand nicht zu finden. Das Laufwerk hat die genaue Bezeichnung TEAC FD55F 31U. Wer weiß darüber etwas?"
- "Wo kann man beim TEAC FN35C die Drehgeschwindigkeit abgleichen?"

Leider kennen wir keines dieser Laufwerke aus eigener Anschauung - kann jemand helfen? Danke!

Billiglösung

Viele haben sich - nach der Anleitung der Zeitschrift c't, Ausgabe 2/88 - eine billige Platte selbst gebaut; kommerzielle Versionen gibt es auch schon. Zwar finden wir diesen Anschluß nicht besonders glücklich, blockiert er doch den DMA-Bus bis in alle Ewigkeit für andere ACSI-Geräte (Laserdrucker, Wechselplatte, Streamer, CD-ROM...). Trotzdem besteht heftiges Leserinteresse, den Diskmonitor SED und auch die restliche KLEISTER-Software an diese Lösung anzupassen.

Bis auf die Menüs HARD und HDX im SED4.0 sind alle Optionen hardware-unabhängig, laufen also mit der c't-Lösung. Die Optionen in HARD und HDX verlassen sich darauf, daß angeschlossene Platten das ACSI-Protokoll beachten, was die "Billiglösungen" nicht tun. Zudem nutzt der SED Eigenheiten der ATARI-Platten, die etwa ein OMTI-Controller nicht bietet. Eine Anpassung ist also aufwendig, aber wir sind bereit, das in Kooperation mit Lesern anzupacken. Findet sich dazu ein programmierwütiger Besitzer des c't-Adapters, vorzugsweise in unserer Nähe (Unterfranken)?

In eigener Sache

"Anton Stepper ist ein Pseudonym von CB."
FALSCH! Warum glaubt mir nur keiner, daß Anton verdammt echt ist?

"Claus Brod ist ein Pseudonym von Anton Stepper."
Warum vermutet keiner so etwas? Trotzdem FALSCH!

"Viren können auf schreibgeschützte Disketten schreiben."
FALSCH! Das ist und bleibt eine Ente.

Soviel für diesmal. KLEISTER-Leser und andere: Werden Sie Ihre Fragen los, schreiben Sie an die Redaktionsadresse, zu meinen Händen, oder direkt an Claus Brod, Am Felsenkeller 2, D-8772 Marktheidenfeld (geht schneller). Wenn Sie keine Veröffentlichung wünschen, merken Sie das bitte an.

CB

*

Desktop.inf-Frage & Sprachenvielfalt

Die folgende Bemerkung bezieht sich auf den Artikel 'Wie man mehr aus seinem *Desktop.inf* macht' aus Heft 4/89. Sie schreiben dort: *"Dem '#c' folgen zunächst 16 Ziffergruppen aus je 3 Ziffern."* ... *"Im Monochrommodus ist nur die erste Dreiergruppe von Bedeutung. Dabei steht '777' für die normale und '000' für die invertierende Darstellung."* Da ich über wenig Programmiersprachenkenntnisse verfüge, habe ich mich mit einer Textverarbeitung über die Datei *Desktop.inf* hergemacht und die angesprochene Dreiergruppe geändert. Weder nach einem Reset noch nach einem Kaltstart war das Bild invertiert. Es klappte dann nach Ändern der beiden ersten Dreiergruppen in '000' und '777'.

Obwohl ich vieles noch nicht verstehe, lese ich ihre Zeitschrift mit großem Interesse, ärgere mich aber mit schöner Regelmäßigkeit darüber, daß Probleme und Listings in so vielen verschiedenen Sprachen (C, Assembler, Pascal, Modula 2 etc.) behandelt und veröffentlicht werden. Allein finanziell übersteigt es meine Möglichkeiten, für die nötigen Voraussetzungen zu sorgen, um an ein Umsetzen der Probleme auf meinen Rechner denken zu können (einige Hundert bis Tausend Mark für diverse Compiler und Literatur). Ich kann mir nicht vorstellen, daß ich mit diesem Problem alleine bin, und möchte Sie daher fragen, ob es nicht möglich ist, die Listings mit so etwas wie dem MCI [Anmerkung der Redaktion: Eine (eventuell gepackte) Zahlenmenge, die den Code des Programms darstellt - 'DATA-Zeilen' - mit Fehlererkennung aufgrund von zum Beispiel Quersummenberechnung] zu veröffentlichen. Irgendein kluger Kopf kann doch sicher etwas Vergleichbares für Ihre Zeitschrift entwickeln. Von der etwaigen Lösung, die veröffentlichten Programme auf einer Diskette anzubieten, halte ich nichts, da nicht alle Programme für jeden brauchbar sind und eine Diskette mit ein oder zwei für mich brauchbaren Programmen für 20 bis 30 DM plus Heftpreis wieder zu teuer wären.

H. Kreimeyer, Oldenburg

Red.: Sie haben Recht: In dem Artikel ist es etwas ungenau ausgedrückt: Zu dem Monochrommodus gehören die zwei

ersten Farbregister, so daß auch beide im *Desktop.inf* so geändert werden müssen, wie sie es beschreiben. Man möge uns diese kleine Ungenauigkeit verzeihen.

Nun zu Ihrer Anregung mit den Programmdaten-'Listings'. Da uns schon mehrere Leser bezüglich dieses Themas geschrieben haben, möchten wir an dieser Stelle einmal Stellung nehmen. Sicherlich haben wir uns schon oft Gedanken darüber gemacht. Leider ist der ATARI ST ein sehr sprachbegabter Rechner, dessen Fremdsprachenkenntnisse sich immer mehr ausweiten. Angefangen hat es mit C, Assembler, BASIC und Pascal; Fortran, Modula 2, und viele andere folgten. Wem soll man es recht machen? Zugegeben, C ist die 'Muttersprache' des ST. Daher ist es auch nicht verwunderlich, daß viele Listings in C veröffentlicht werden.

Um das Dilemma ein wenig zu umgehen, haben wir uns entschlossen, mehr Programme in OMIKRON.BASIC zu bringen, da man bei dieser Sprache davon ausgehen kann, daß sie jeder, der einen ST hat, auch besitzt - sie ist als Sprache beim Kauf dabei oder ansonsten für einen sehr geringen Preis zu beziehen. Von der von Ihnen angesprochenen Lösung der 'Datazeilen', also dem Abdrucken der eigentlichen Zahlen des Programmes, halten wir bei einem Rechner wie dem ATARI ST relativ wenig. Ein einigermaßen interessantes und verständliches Programm wird durch das Kompilieren inklusive Initialisierung derart lang, daß ein Eintippen der Zahlen nicht zu verantworten wäre (oder würden Sie 2 kBytes, was teilweise noch sehr untertrieben ist, Zahl für Zahl eintippen?). Auf der anderen Seite wurde uns bei ähnlichen Zahlenkolonnen schon ab und zu Seitenschinderei vorgeworfen, was auch irgendwo verständlich ist. Nebenbei, im folgenden Leserbrief finden Sie ein Listing, das kompiliert 6803 Bytes ergeben hat - der ST ist halt nicht sparsam bei Bytes. Die Methode der Programmcodeveröffentlichung hätte nur bei reinen Assemblerprogrammen Sinn, da hier die Zahlen im Vergleich zum eigentlichen Assemblertext noch kurz sind, aber trotzdem würden wir immer auch den Source-Text abdrucken. Langer Rede kurzer Sinn: Momentan sehen wir keine vernünftige Alternative zum reinen Abdrucken des Quelltextes und Herausgeben einer Monatsdiskette. Wir versuchen aber, einen großen Teil unserer Leser anzusprechen und somit auch ab zu mal eine nicht so bekannte Programmiersprache zu bringen. Einen kleinen Trost und Tip: Versuchen Sie mal, die C-, Modula 2- oder etc. -Programme zu lesen und in die Sprache Ihrer Wahl zu übersetzen. Sie werden merken, wie oft dies gut geht und daß Sie dabei viel lernen.

*

Druckerausgabe in C

Seit einem etwa dreiviertel Jahr lerne ich als Autodidakt die Sprache C und komme mit ihr ganz gut zurecht. Ich vermisse aber in allen Lehrbüchern eine meiner Meinung nach auch für Anfänger sehr wichtige Erläuterung. Ein Buch zum Beispiel beschreibt auf ein paar Seiten sehr gut, wie man "C'est si bon" auf den Bildschirm ausgeben kann. Aber wie bekomme ich diesen schlichten Satz mit meinem Drucker auf ein Stück Papier? Darauf geht das Buch mit keiner Zeile ein. Das gilt auch für andere Bücher! Im C-Kurs in den ST-Computer-Heften 5/86 bis 9/86 ebenfalls Fehlanzeige. Das Handbuch für Drucker gibt auch nur Beispiele in BASIC mit LPRINT.

Für mich ist es lästig, daß ich - ich bin Hobby-Mathematiker - Ergebnisse mit 13 Nachkommastellen vom Bildschirm abschreiben muß. Meine Frage lautet daher: Mit welcher Befehlssequenz kann ich in der Sprache C meinen Drucker veranlassen, ein Wort auszudrucken?

H. Möller, Frankfurt

Red.: Die Tatsache, daß dies in den Büchern nicht erwähnt ist, ist wohl darin begründet, daß der Drucker nicht mehr zu STDIO (Standardein-/ausgabe) gehört, sondern hardwarespezifisch ist. Die erste Methode, die Sie im folgenden finden, funktioniert so, daß Sie Zahlenausgabe in einen String mit *sprintf()* umleiten und dann mit einer eigenen Routine *prn_out()* den String zeichenweise ausgeben.

```
/************************************************/
/* Druckerausgabe: hardwarespezifisch direkt    */
/* übers BIOS -> Cprnout()                      */
/************************************************/

#include <stdio.h>      /* Standard-I/O */
#include <osbind.h>     /* BIOS-Routinen */

void prn_out(char*);     /* Funktionsdefinition */

void main()
{
   int a=1;          /* eine Variable */
   float f=3.141592654;  /* noch eine */
   char mystring[20];    /* Ausgabestring */
    sprintf(mystring,"%d %3.7f\n",a,f); /* Ausgabe in
String */
   prn_out(mystring);    /* Druckerausgabe */
}

void prn_out(string)     /* String als Übergabe */
char *string;
{
   int i=0;          /* Zähler*/

   while(string[i])      /* bis zum Ende */
     Cprnout(string[i++]); /* Zeichen ausgeben */
}
```

Sicherlich wird man bei einem anderen Rechner aber keine Routine *Cprnout* finden. Die einzige Methode, die ich noch herausgefunden habe, ist die auf dem ANSI-Standard beruhende folgende Art. Sie ist in Turbo-C geschrieben und müßte daher auf allen ANSI-Cs funktionieren. Mit Laser-C funktioniert es jedenfalls nicht (hoffentlich bald, denn nach dem Source-Level-Debugger müßte ja Zeit für ANSI-C vorhanden sein)! In diesem Fall ist es möglich, eine Datei auf den Drucker zu öffnen, da im neuen Standard eine Datei mit Namen 'prn:', 'com:' und 'aux:' direkt auf bestimmte Ausgabegeräte umgelenkt und nicht mit diesem Namen auf einem Laufwerk angelegt wird.

```
#include <stdio.h>    /* Standard-Ein/Ausgabe */
#include <string.h>   /* Stringbefehle */

void main()
{
   FILE *fd2;          /* Datei-Handle */
   int a=1;             /* 'ne Variable */
   float f=3.141592654;    /* PI */
   char mystring[20];     /* String zum Ausgeben */

   sprintf(mystring,"%d %3.7f\n",a,f); /* in den String
*/
   fd2=fopen("prn:","w");          /* PRN: ist Drucker */
   fwrite(mystring,sizeof(char),strlen(mystring),fd2);
/* Ausgabe */
   fclose(fd2);              /* Datei schließen */
}
```

*

Speicherzugriff in C

Ich möchte in C einige der geschützten und ungeschützten Adressen (auch Bitvektoren) manipulieren. Die Poke-Befehle arbeiten im Supervisor-Modus nicht einwandfrei. Wie kann ich die Vektoren und Adressen beeinflussen? Wie werden Daten an einer bestimmten Adresse gespeichert?

Maximilian Menig, Erbach

Red.: Gerade C bietet sehr gute Möglichkeiten im Verändern von Adressen. Prinzipiell geht man folgendermaßen vor: Man definiert einen Zeiger auf den Datentyp (char, int...), auf den man zugreifen möchte, und weist diesem Zeiger über eine CAST-Operation (Erzwingen eines Typs) den Wert der Adresse zu, auf die man zugreifen möchte. Dann kann durch '*zeiger' in den Zeiger etwas hineingeschrieben oder aus ihm herausgelesen werden. Möchten Sie auf geschützte Adressen zugreifen, muß sich der Prozessor im Supervisor-Modus befinden, was über den XBIOS-Befehl Super() möglich ist. Dieser wechselt durch Übergabe des Wertes 0L in den Supervisor-Modus und gibt den aktuellen Stapelzeiger zurück. Der Supervisor-Modus wird durch 'Super(stapelzeiger)' wieder verlassen, wobei das Argument den alten Stapelzeiger wieder herstellt - deshalb hat man sich den alten Wert vom ersten Aufruf zu merken. Um Ihnen zu zeigen, wie dies praktisch geschieht, folgt dieser Erklärung ein kleines Listing, das in MEGAMAX Laser C erstellt wurde:

```
/* Megamax Laser C - Zugriff auf (reservierte)
   Speicherbereiche */

#include <osbind.h>

#define PHYSTOP  0x42eL      /* Systemvariablen */
#define SEEKRATE 0x43eL
#define SSHIFTMD 0x44cL

main()
{
   char *poi_1, val_1;       /* Zeiger auf 1 Byte   */
   int  *poi_2, val_2;       /* Zeiger auf 2 Bytes,
                                ein Wort */
   long *poi_4, val_4;       /* Zeiger auf 4 Bytes,
                                ein Langwort */
   long  ssp;                /* alter Stapelzeiger */

    poi_1 = (char*) SSHIFTMD; /* Adreßzuweisung mit
                                 CAST-Operation */
    poi_2 = (int*)  SEEKRATE;
    poi_4 = (long*) PHYSTOP;

    ssp=Super(0L);        /* In Supervisor-Modus schalten */
    val_1= *poi_1;
    val_2= *poi_2;
    val_4= *poi_4;
    Super(ssp);           /* Zurückschalten */

      printf("Modus: %d  Seekrate: %d   Phystop: %lx\n",
          val_1, val_2, val_4);
   Crawcin();             /* Auf Taste warten */
}
```

Festplatte und das RAM-TOS?

Ich habe folgendes Problem: Ich besitze seit zwei Jahren einen 260ST, der inzwischen auf 1 Megabyte aufgerüstet ist, und habe zu Weihnachten die Festplatte Megafile 20 bekommen. Im Handbuch der Festplatte war zu lesen, daß man das TOS im ROM haben muß, um die Festplatte ordnungsgemäß betreiben zu können. Da bei meiner Festplatte die Treiberdiskette fehlte, habe ich mir von einem Bekannten die Treibersoftware für die VORTEX HD Plus besorgt. Mit dieser Software ist es mir gelungen, obwohl ich das TOS von der Diskette lade, die Festplatte zu formatieren und zu bespielen. Bis jetzt habe ich keine Fehlfunktion bemerkt. Nun frage ich mich natürlich, welchen Sinn hat dann noch das TOS im ROM, und kann ich meine Festplatte kaputtmachen, wenn ich meinen Rechner nicht umrüsten lasse?

Christian Hartwig, Hamburg

Red.: Ihre Festplatte können Sie, bezogen auf die Hardware, sicherlich nicht kaputtmachen, allerdings besitzt das RAM-TOS des ATARI ST an einigen Stellen noch Fehler, die sich beim Betrieb der Harddisk in Datenverlust bemerkbar machen können. Das bekannteste Problem ist, daß die Festplatte bei einer großen Anzahl von Ordnern auf einer Partition nicht mehr das tut, was sie soll. Der ATARI-Treiber schließt übrigens das RAM-TOS schlichtweg aus, in dem er beim Start abfragt, ob das ROM-TOS vorhanden ist - ist dies nicht Fall, so installiert er sich erst gar nicht. Zusammengefaßt kann man sagen, daß es sicherlich sinnvoll ist, die Festplatte nur mit ROM-TOS zu fahren. Wenn Sie möchten, können Sie auch warten, denn bald bringt ATARI das neueste TOS auf den Markt, das auch kompatibel zum neuen TT sein soll.

*

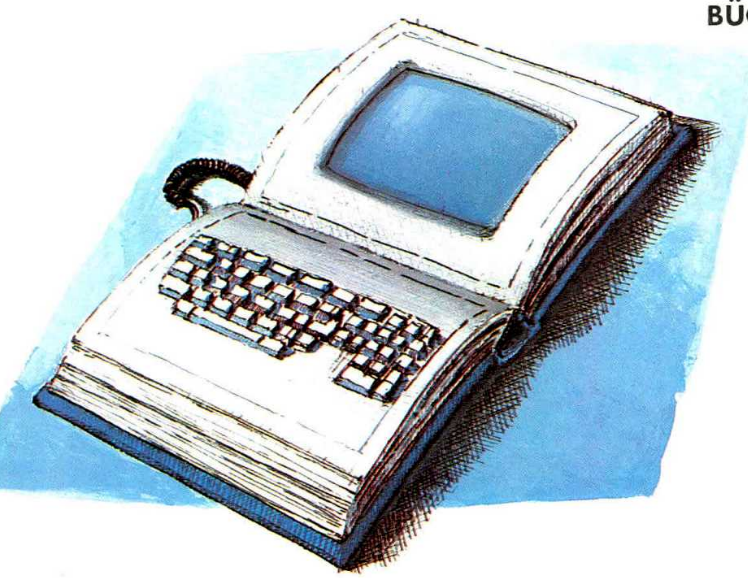
SIGNUM-24-Nadeldruckertreiber für OKIDATA Microline 390

Ich bin Besitzer eines OKIDATA Microline 390, der, ebenso wie der NEC-P6, einen Zeilenvorschub von 1/360 Inch kennt, jedoch dafür einen anderen Steuercode als der P6 benutzt. Deshalb war bisher der Ausdruck über 360x360 im SIGNUM!2-Druckertreiber nicht möglich. Ich habe nun die verantwortlichen Adressen im 24-Nadeltreiber herausgesucht und entsprechend gepatcht...
Der Steuercode, der vorher für den P6 gesendet wurde, war hex 1C 33 und ist für den OKIDATA in hex 1B 5B umzuwandeln. Das BASIC-Programm, welches den PATCH-Vorgang ausführt, sieht folgendermaßen aus:

```
OPEN "I",#1,"A:\SIGNUM\PR24N.PRG"  !Druckertreiberfile
                                     öffnen
SEEK #1, 10875       !Zeiger auf 1. Adresse positionieren
OUT  #1, 27          !1. Steuercode patchen
SEEK #1, 10885       !Zeiger auf 2. Adresse positionieren
OUT  #1,91           !2. Steuercode patchen
CLOSE #1
```

In der ersten Zeile ist der Pfad des Druckertreibers einzutragen. Sollte Ihr Drucker einen anderen Code verwenden, können Sie dies in der 3. und 5. Zeile ändern. Da ich nicht beurteilen kann, ob es verschiedene Druckertreiberversionen gibt, und weil unter Umständen beim Patchen etwas schiefgehen kann, sollte man vor dem Patchvorgang eine Sicherheitskopie anfertigen und mit dieser arbeiten. Damit dürfte einem noch besseren Ausdruck nichts mehr im Wege stehen.

Anmerkung der Redaktion: Machen Sie sich vor dem Patchversuch auf jeden Fall eine Sicherheitskopie des Druckertreibers. Sollte Ihnen der Patch nicht gelingen, wenden Sie sich an Application Systems /// Heidelberg.



Peter Haberäcker

Digitale Bildverarbeitung

Grundlagen und Anwendungen
2. Auflage, 1987
Carl Hanser Verlag, München Wien
377 Seiten
ISBN 3-446-14901-5

In einer der letzten Ausgaben wurde im Rahmen des Imagic-Grafikkurses schon kurz darauf eingegangen; die ganze Informatikwelt redet seit einigen Jahren mit steigendem Interesse davon: das Thema heißt 'Digitale Bildverarbeitung (image processing)'. Im Bereich der digitalen Bildverarbeitung schießen besonders im englischsprachigen Raum die Bücher wie Pilze aus dem Boden. Meine Erfahrung dahingehend ist, daß die Bücher entweder hoffnungslos veraltet oder mit höherer Mathematik verseucht sind, was dem Einarbeiten in diesen sehr interessanten Themenbereich nicht gerade förderlich ist.

Umso mehr ist es erfreulich, daß ich Ihnen ein deutschsprachiges Buch vorstellen kann, das nicht allzusehr in mathematische Höhenfluge ausartet und trotzdem eine relative gute Übersicht und Einführung in das Themengebiet darstellt, was wahrscheinlich daran liegt, daß der Autor Dozent an der Fachhochschule München ist und Ahnung von Didaktik hat. Es sei trotzdem darauf hingewiesen, daß weitergehende Kenntnisse von Summen- und Matrizenberechnungen vorhanden sein sollten.

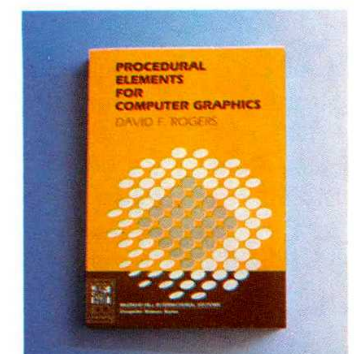
Eine Aufzählung aller Kapitel zu geben, wäre hier zu umfangreich, trotzdem sollen ein paar Kapitel stellvertretend sein: Digitalisierung, Charakterisierung und Speicherung von Bilddaten, Grauwertverteilung (Histogramm, Äquidensiten), Operationen im Orts- und Frequenzbereich (Konvolutionen), Mustercharakterisierung und -erkennung und vieles mehr. Einen vollständigen Abriß über dieses Thema zu geben, ist absolut unmöglich, so daß die beiden Hauptthemenbereiche Bildaufbereitung (image enhancement) sowie Mustererkennung (pattern recognition) durch die Kapitel recht gut abgedeckt sind.

Am besten gefällt mir, daß die Themen auch ausprogrammiert in PASCAL vorliegen, so daß so mancher Programmierer die Theorie aufgrund des Algorithmus' besser verstehen und sehr schnell in die Praxis umsetzen kann. Diese Programme können sogar als Diskette beim Autor bestellt werden. Wem dies nicht ausreicht, dem sei gesagt, daß sich an die Kapitel eine reiche Sammlung von Übungen anschließt. Sehr positv ist, daß man die Lösungen der Aufgaben in Kurzform im Anhang findet. Damit die Lösung der Aufgaben nicht zu schwer wird und man bestimmte Themen besser aufarbeiten kann, findet sich am Ende eines jeden Kapitels eine umfangreiche Auswahl an Sekundärliteratur. Einziger Kritikpunkt, den aber praktisch 95% aller Bücher dieses Themengebiets betreffen, ist, daß ab und zu der Praxisbezug ein wenig fehlt, so daß eine Wertung oder der Einsatz der einzelnen Verfahren sehr zum Verständnis beitragen würde. Auch wäre eine ausführlichere Behandlung von verschiedenen Themen wie Kontrastanhebung über Lookup-Tabellen (histogramm-stretching) nicht schlecht gewesen. Außerdem fehlt eine Abhandlung über die Funktion und Anwendung von heutzutage recht oft eingesetzten Zeilenkameras.

Trotzdem hält man mit diesem Buch ein Werk in der Hand, welches bei intensivem Lesen einen guten Einblick in das Thema digitale Bildverarbeitung gibt und den ein oder anderen sicherlich in die Lage versetzen wird, Probleme aus der Praxis besser abschätzen zu können. Denn nicht immer ist es sinnvoll, teure Bildverarbeitung einzusetzen, wenn vielleicht optische Sensoren auch zu dem gleichen Ergebnis führen.

SH





David F. Rogers

Procedural Elements for Computer Graphics

4. Auflage 1988 (1985)
McGraw-Hill International Editions
433 Seiten
ISBN 0-07-053534-5

Gehören Sie auch zu denen, die fasziniert auf Computergrafiken schauen und sich immer wieder fragen, was eigentlich dahinter steckt? Oft habe ich mich gefragt, wo man Algorithmen herbekommt, mit denen man auf schnelle Weise Linien und Kreise zeichnet, bei dreidimensionalen Objekten verdeckte Kanten beachtet und verwertet oder Schattierungen (Shading) an Objekten berechnet. Heutzutage findet man viele Bücher über diese Themen, die aber den Nachteil haben, daß sie sich stark von der Praxis lösen und vor lauter Mathematik den Kern der Sache verdecken - der Algorithmus muß dann auf Grundlage dieser Mathematik selbst entwickelt werden.

Auch wenn man vielleicht denken könnte, ein Buch von 1985 sei schon veraltet, so trifft es für dieses nicht zu. Das Buch gliedert sich in fünf Kapitel, in denen über die Hardware von Grafikein- und -ausgabegeräten, Algorithmen zur pixelorientierten Grafikausgabe, Clipping, verdeckte Kanten und Flächen sowie Schattierungsalgorithmen informiert wird.

In diesem Buch habe ich zum ersten Mal eine verständliche Erklärung des Bresenham-Linien-Algorithmus' gefunden,

den ich wie den im Buch befindlichen Kreis-Algorithmus ohne Probleme implementieren konnte. Wenn auch meines Erachtens das Kapitel über Clipping (fensterorientierte Grafikausgabe) etwas lang geraten ist, wird man in den folgenden Kapiteln über versteckte Kanten und Flächen sowie die Oberflächengestaltung (rendering, shading, raytracing) wieder mit interessanter Materie fundiert beraten, wobei es hier natürlich nicht ohne Mathematik (Abiturwissen und darüberhinaus) geht.

Hervorzuheben an diesem Buch ist die Tatsache, daß zu fast jedem Thema ein Algorithmus im Pseudocode (leicht in eine beliebige prozedurale Programmiersprache übertragbar) oder Flußdiagramm angegeben ist, der durchdacht ist, (nach meiner Erfahrung) funktioniert und (teilweise an Beispielen) erklärt wird. Das Ganze wird durch viele Tabellen, Grafiken und z.T. farbige Bilder unterstützt, was dem Verständnis dieser teilweise abstrakten Materie sehr förderlich ist. Möchte man sich darüberhinaus Wissen aneignen, so findet man hinter jedem Kapitel einen umfangreichen Literaturverweis, auf den schon während der Kapitel hingewiesen wird.

Am Schluß des Buches stehen ein relativ ausführlicher Index sowie eine Sammlung von Aufgaben, die der Leser gestellt bekommt. Diese Aufgaben dienen der Vertiefung von Teilgebieten des Buches und sollen als Projekte bearbeitet werden, so daß sie sicherlich als sehr interessante Studienarbeiten zu verwenden sind. Leider gibt es zu diesen Projekten keine Lösungen, die in diesem Buch oder als Erweiterung angeboten wären. Ein Nachteil, den man schon am Titel erkennt, soll aber nicht verschwiegen werden: Wie viele Bücher aus dem Bereich der Computer ist auch dieses in englischer Sprache verfaßt und soweit mir bekannt, nicht in deutscher Sprache verfügbar. Wen diese Sprachbarriere allerdings nicht abschreckt und wer einmal in die Grundlagen der Computergrafik einsteigen möchte, dem sei dieses Buch wärmstens empfohlen.

SH

Kleinanzeigen

BIETE HARDWARE

1 ATARI ST4 Computer compl. m. Blitter + Mon. SM124 + Laserprinter SLM804 + Laserbrain neuw. DM 3985,- 06151/25766

520ST+, Drucker LC10, 2 Laufwerke Monitor SM124 incl. Zubehör **komplett** zu verk. **VB: 1800,- DM** Tel. 05373/6829 (Heucke) n. 18 Uhr

---

Atari 520STM SF354 Maus Monitor Top günstig 09543/7781 oder 3301

RHO-Bus, PC-Gehäuse, 2PiO-Karten 1040STF, SM124 Neu W. Bosse, Tel. 0228/616606

Monitor SM124 1 Jahr alt VB DM 300. Tel. 18h 02683/32621

Verk. Speichererw. ☎: 0431/569216

HF-Modulator, 2 Monate alt, max. 20h gelaufen, VB 150 DM Tel. 02271/61357

Atari ST Blitter-ROM-TOS V1.4 Fast-Harddisk, auch mit IBM-Zeichensatz, 100,- DM, 02630/7525

Neue Mega ST-Tastaturen kompl. preisgünstig Tel. 08142/51003

Blitter-TOS o. altes TOS o. TOS 1.4 NEU gleichzeitig in allen ST, umfangr. Anleit. K. Ratsch, Herner Str. 127, 4350 Recklinghausen, 02361/28442 ab 19.00 Uhr

---

BTX-Computerinterface alle PC/HC, DM 99, Info DM 1, Han72600309

Digitaltemperaturmessmodul mit Softwareauswertung 78 DM – Star LC10 569 DM – NEC P6+ 1444 DM – Neu – Suche Laser C. Tel. 08453/2585

Gelegenheit! Test ST-Comp. 5/89 Orig. „SPAT" Scanner Flachbett DIN A4 200dpi s/w + grau 16 Grst. incl. Thermodr./Kopierer Softw. **nur DM 1098,-** tägl. 09445/1486

---

5 1/4 Zoll Teaclaufw. mit A/B u. 40/80 Track Umsch. incl. 50 Disketten 300 DM 0202/660422

Kabel ST → P2200 gepuffert 1,8 m DM 35,- Normal: 15 DM **Farbband P2200: 11,95 DM** bei: Horst Braun, Mollwitzstr. 10 5000 Köln 60, Tel. 0221/741489

Atari 1040 in AT Gehäuse 20 MB 2 Floppys, PC Ditto, First Word, DTP, div. Progr. VB 2300 DM 08142/16283 ab 18 Uhr wg. Systemwechsel

---

Atari Laserdrucker SLM 804 VB 2222 DM Festplatte für Atari ST 100 MB VB 2000 DM werktags ab 16⁰⁰ Telefon: 0511/814098

Verk. Speichererw. T: 0431/569216

Atari ST-Blitter-ROM-TOS V1.4 (neu), auch mit IBM-Zeichensatz, FAST-Harddisk, 100,- 02630/7525

Verk. MegaST, SM124, 3 1/2" Flop., Uhr, div. SW, 2000,- Tel. 0525465441

BIETE SOFTWARE

■■ **Brandneue PD-Software** ■■
■ **selbstverständlich Virenfrei** ■
■ Fast täglich neue Programme, ■
■ direkt vom Autor, die es nur ■
■ bei mir gibt! Riesiges Ange- ■
■ bot auf über 130-Seiten! **Alle** ■
■ **Programme sind einzeln aus-** ■
■ **wählbar!!!** Der absolute Hit: ■
■■■ nur 1,5 Pf pro KByte ■■■
■ Sie können **eigene** Disketten ■
■ senden oder auf **Markendisks** ■
■■■■■ bestellen!! ■■■■■
■ Bearbeitung erfolgt selbst- ■
■ verständlich **postwendend!!!!** ■
■ Kataloganforderung gegen ■
■ eine Schutzgebühr 5,- DM, ■
■ Anrechnung bei Kauf ■
■ **ST Profi-Partner** ■
Mönkhofer Weg 126, 24 Lübeck 1
■■ Telefon: 0451-505367 ■

NEU ★ ★ NEU ★ ★ NEU ★ ★ N
EDAS komf. **Notenverwprg.** fürE
U alle Lehrer! Ausführliche U
N Anleitung! Beg. Bedienung! N
E Info geg. fr. Rückumschlag E
UHJ. Merkel/Nahestr. 28/66 SbrU

PUBLIC-DOMAIN-SOFTWARE ST-DISK. je DM 4,90 Versand Scheck 3,00, Nachn. 5,50 W. Schlude, Dorfstr. 37 7488 Stetten a.k.M. 1

ST-Pascal+ 120, **Lattice C** 100, **MCC-Ass.** 80, **Adimens** 90, **1stWord+** 90, **Hanse** 30, **Profimat** 40, **Starcomm** 30, Tel. 09191/32623

■■■■ **PD-VERSAND** ■■■■
■ Katalogdisk + 7 Prg's für 6 DM ■
■ (Briefm.) bei: Stefan Heigert ■
■ Berndesallee6, 6501 Heidesheim ■
■■■■■■■■■■■■■■■■■■

STARWRITER orig. DM 100, Tel. 069/880257

GfA/Accessorie Maker. Incl. komfortabler Shell 07634-2195

★ ★ ★ **NOTDATMAN** ★ ★ ★
Noten/Daten-Manager für Lehrer
Komfortable Notenverwaltung
Auswertung v. Korrekturlisten
Schülerdatei, Absenzenlisten
zahlr. Statistikfunktionen
INFO g. Freiumschlag (1,70)
M. Groh, Zedernstr. 29, 8510 Fürth

■ ■ **Public Domain** ■ ■
■ ■ **Software** ■ ■
■■ „nicht jeder ■■
■ ■ Shit zählt..." ■ ■
■ ■ ■ ■

Endlich fertig: Unser neuer Gratiskatalog mit 850 Disks. Jetzt bringt PD-Soft noch mehr Spaß, denn jetzt gibt es, nur bei uns, die Zeitschrift PD-ACTION! Eine Fülle von Berichten, Sensationen und Tests werden Sie begeistern. Fordern Sie daher unseren Gratiskatalog oder gegen DM 5,– den Katalog mit der aktuellsten PD-ACTION an!! Sählen Sie nicht unter -zig Anbietern, nehmen Sie die Nr. 1 – Wir lassen die Bits wackeln!
Klaus Kohler
Don-Carlos-Str. 33 B
7 Stuttgart 80

Wärmebedarf DIN4701 + K-Zahl ★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder, Eichendorffstr. 15 · 5030 Hürth

---

The Best of Public Domain
Katalog + Probedisk DM 5,- Reinh. Rückemann, Grundstr.63 56Wpt.22

Verkaufe Timeworks DESKTOP PUBLISHER ST Original mit Handbuch zu SFr. 200,– Ochsner Sport, Frl. Ingold, CH-8050 Zürich, Tel. 01 3113035

Unbenutztes **Megapaint II jr** zu verk. DM 75,- Tel. 089/4304776

PD-SOFTWARE
Alle ST-Disks SS je DM 4,– (incl. 2DD-Diskette!) Doppelseitig DS je DM 6,– Versand: bei Scheck DM 3,– bei Nachnahme DM 8,– N. Twardoch, Gröchteweg 22 4902 Bad Salzuflen 1

Superbase Professional Original (neu) mit Registrierkarte wegen Systemwechsel Tel. 06142/42036 DM 479,–

★ **PD für ATARI-ST, IBM** ★ **Atari**(500), **IBM**(2500), 3–6 DM Kat: PC-10DM (7 Disk.), ST-5DM Johrend, Neusalzer Str. 9, 85 Nbg

---

Lohn-Einkommensteuer, Miet-Lastenzuschuß, Rentenber./Beamtenversorgung.
H-I-SOFTWARE
Niederfelderstraße 44
8072 Manching · Tel. 08459-1669

Public-Domain-Blitzversand! Riesenauswahl! Preise: ab DM 4,– einseitige Disketten ab DM 6,– doppelseitige Inclusive Diskette!!! Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar! Gratisliste anfordern bei: A. Gauger Software Buhlstraße 16a, 7505 Ettlingen 0 72 43 / 3 18 28 Bitte Computertyp angeben!!

■■ **Public-Domain-Software** ■■ Gratiskatalog bei: O. Schwede, Röntgenweg 9/1, 7050 Waiblingen

---

Spectre <–> <–> ALADIN Konv. von Mac Paint/Ascii/Font Dateien ★ PRG für DM 60 Info bei A. Hekermans Billertstr. 17, 7000 Stuttgart 80

Wärmebedarfsberechnung DIN4701 KZahlberechnung DIN4108 Dampfdiffusion-Tauwasserschutz Wärmeschutznachweis WäSchV Dipl. Ing. V. Koch, Am Mehnacker 11 3563 Dautphetal 3, Tel. 064687652

---

soft-STation
2 PD's von ST-Comp.
Fortl. Nr. (1-2 usw.) DM 5,-
Bel. Komb. DM 7,-
Neue 2000er Serie DM 7,–
Serie Zeitschr. Public Domain DM 7,-
Disketten wahlw. blau, weiß, rot, grün, gelb od. orange
Verp. Porto plus NN DM 7,-
Tel. 07195-53707

Originale: LATTICE C DM 140,- Megam. Laser DM 160,- spc Modula DM 190,- Becker Datab. DM 30 Tel. 06073-62315 nach 20 Uhr

---

Verkaufe komplette PD-Sammlung der ST-Computer. **Alle** Updates Preis: VB Tel. 0231-270310

---

verschiedene **Calamus-Fonts, kostenlose** Info anfordern bei: A. Schreiber, Flachterstr. 32, 7251 Weissach, **Tel. 07044/32834**

HP-format. 3,5" Disketten am ST lesen/beschreiben. Info: Dr. M. Poech 0711/297149 (18–22h).

PLOT_ST für Atari-Fortran
Einfache Ausgabe grafischer Darstellungen auf Bildschirm und Drucker für 95,– DM. Kompatibel mit Großrechner Standard (Plot–79). Handbuch DM 15,– (wird verrechnet); Info gratis. PLOT-ST sowie weitere Fortran-Tools bei: **M. Gamer,** Friedrichsring 26, 6050 Offenbach

---

Original-Software Laufend neue Programme a. Lager Gratiskatalog anf. o. anrufen Wünsch & Dauler Soft- u. Hardware, 07231/766595 Friedenstr. 212, 7530 Pforzheim

---

■■ LIGA-TAB 2.0 ■■
■■ Sport-Tab.-Verwaltung ■■
■■ Info gegen frank. Rück- ■■
■■ umschlag oder gleich be- ■■
■■ stellen für 40,– DM bei: ■■
■■ T. Eckhardt, Fr.-Ebert- ■■
■■ straße 1, 6306 Langgöns ■■

★ ★ ★ aladin ★ ★ ★
★ ★ Filetransfer nach TOS ★ ★
mit Konvertierung von Text und Grafik nach Degas, STAD und IMG 59 DM, M. Nieß, Am Kottenberg 34, 4799 Borchen, Tel. 05251/391849

ADIMENS 2.3, mit Handbuch, Praxis-Buch u. Übungsdisk. DM 20, Tel. 069/880257

Gebrauchte Original-Programme Liste: Telefon 06721/43575

PD-Sammlung von Grafik + DTP mit Druckunterlagen. Tel. 06721/43575

GROSSE GRAFIKSAMMLUNG! ca. 2000 IMG-Grafiken (z. B. für Wordplus) DM 25,– 06302-3338

■ ■ **Public Domain** ■ ■
■ ■ **Software** ■ ■
■■ für alle ■■
■ ■ Atari ST-Modelle ■ ■
■ ■ ■ ■

Bei uns gibt es die PD-Soft, die jedem Spaß macht! Gratiskatalog anfordern!!
Klaus Kohler
Don-Carlos-Str. 33 B
7 Stuttgart 80

TIM BUCHFÜHRUNG DM 150,- Publishing Partner DM 125,- Tel. 06074/90378 (alles Org.)

Ringbücher Buchschuber Zubehör und Disketten-Versandtaschen SMS Siegmund, Gartenstr. 24 3051 Suthfeld, Tel. 05723-81415

PD-Software ab 2,- DM/Diskette 4 komplette PD-Nummern auf einer Diskette für 8,- DM Tel. 02721/2432 von 12-21 Uhr

SUCHE HARDWARE

1040STF + SM124 (0212) 208613

Einzelblatteinzug für NEC P6 alt gesucht (089) 30622485/1679047

MEGA 2 ODER 4 GESUCHT WERNER 0221/7409422

SUCHE SOFTWARE

Suche Lernprogramm für dt. Rechtschreibung 02241/337114

Suche **fibuMAN** o.ä., **Tabellenklk.** ausf. Anl. zu DBase u. **Wirtschsim.** Lutz Mathes 06172-71160

KONTAKTE

1 PD = 1 DM
– Div. Orig. 06436/3639

VERSCHIEDENES

■■■■■■■■■■■■■■■■■■
■ **CCS COMPUTER SHOP** ■
■ Hard & Software – Ersatzteile ■
■ Markendisketten 3,5 10 St. ■
■ 1D ab 20,- DM 2D ab 25,- DM ■
■ bei gr. Mengen Rabatt. ■
■ Grafiken für Signum/STAD ■
■ ca. 1500 20,- DM ■
■ Info anfordern: CCS Computer ■
■ Shop-Langenhorner Ch.670d ■
■ 2000 Hamburg 62 ■
■ Computer angeben. ■

Speichererw. f. ATARI520,260ST auf 1MB m. Einbau 260 DM. Atari ST auf 2,5MB 860 DM. Bausatz auf 1MB 220 DM, auf 2,5MB 700 DM, Tel. 069/686491

Interaktive Leiterplattenentfl. mit Bauteilebibl. u. Editor, Drucker- u. Plottertr. HPGL Info unter Tel. 02325/70796

Laser-Ausdrucke: (0228) 692035

Suche Mitspieler für Briefsimulationsspiel; Michael-Andreas Tänzer, Krausenstr. 2, 3 Hannover

---

Druckertreiber und Anpassungen, Tools und Downloadfonts für 24-Nadel-Drucker. Günterberg 0221-519950

Suche/Tausche Drucker-Fonts f. Star, Nec, Epson, 0202/640389

Tausche MegaSt, SM124, 3 1/2", Uhr, div. SW geg. PC-AT Tel. 0525465441

Weide ELEKTRONIK

Weide Elektronik GmbH
Regerstraße 34
D-4010 Hilden
Telefon (0 21 03) 4 12 26

Ladenlokal: Gustav-Mahler-Straße im Einkaufszentrum
Telefon Ladenlokal (0 21 03) 3 18 80 · Telefax (0 21 03) 3 18 20
viele Parkplätze direkt am Laden

ATARI System-fachhändler
Leasing für Komplettsysteme möglich
Atari + PC XT/AT

Schweiz
DTZ Data Trade AG
Langstrasse 94
CH-8021 Zürich
Tel. (01) 2 42 80 88
Fax (01) 2 91 05 07

Österreich
Reinhart Temmel GmbH
Markt 109
A-5440 Golling
Tel. (0 62 44) 70 81-0
Telefax 7 18 83

SOFTWARE

WE SOFT
SINUS 79.-
mathem. Software für Schule und Labor

UNISHELL 98.-
GEM-Shell

CRYPT 79.-
sehr schnelles Verschlüsselungs-programm als .ACC, .TTP, .PRG Schützt Ihre Dateien vor unbefug-ter Benutzung! Verdeckte Schlüs-seleingabe, extrem schnell (ver-/entschlüsselt 2,5 MB in ca. 70 Sek. auf der Festplatte), INM.ACE Verschlüsselungen, dynamisches Verschlüsselungsverfahren – äußerst sicher! Als Programm oder Accessory nutzbar.

68881 + WCL 898,-
Ein starkes Gespann
Mit WCL erhalten Sie mit 68881 auf ATARI ST/68000 Systemen die höchstmögliche Rechenge-schwindigkeit. INFO anfordern!

Textverarbeitung

Tempus 2.0	119.-
1st_Word plus	189.-
1st_Extra	89.-
Profi Textmodul	79.-
1st_Proportional	89.-
1st_Adress	145.-
Daily Mail	179.-
Signum 2	398.-

DTP

Calamus, neu, viel schneller	798,-
Outline Art	348.-
Fonteditor	198.-
Vektor Art Graf. Biblioth.	198.-
CG Fonts	a. A.
MegaFonts	a. A.
Das große CALAMUS Buch	39.-
Desktop Knigge (TeWi)	79.-
Desktop Publishing mit Mikrocomputern (Sybex)	59.-
Scan-Service	a. A.
Scanner, versch. Typen 200 – 400 DPI	a. A.

Tabellenkalkulation

LOGISTIX	398.-
LDW PowerCalc	219.-
BASiCALC	78.-
Querdruck f. Tabellenkalkulation	58.-

Bücher

ATARI ST Profibuch Sybex, 988 S.	69.-
Motorola 68000	59.-
Motorola 68020	79.-
Motorola 68030	98.-
Motorola 68881	79.-
Das große OMIKRON BASIC Buch	49.-
Programmieren in Maschinensprache (Sybex)	48.-
GfA Buch TCS + GER	a. A.

Programmiersprachen

Omicron Assembler	99.-
GFA Assembler	149.-
Turbo C Assembler/Debugger	189.-
Omicron Basic Compiler	179.-
Omicron Basic 68881 Compiler	229.-
Omikron Basic Libraries (Midi, Statistik, Numerik, Complex, Easygem)	je 79.-
(Finanzmathem., Inlineass.)	je 129.-
Maskeneditor, Link it, Draw	je 79.-
GFA Basic Interpreter 3.0	189.-
GFA Basic 68881 Int. + Comp.	298.-
Aztek C prof.	398.-
Aztek C developer	598.-
Aztek C S. L. Debugger	148.-
Laser C	388.-
Lattice C	248.-
Mark Williams C	298.-
Prospero C	248.-
Turbo C	189.-
Turbo C + Assembl./Debugger	279.-
AC Fortran	448.-
Prospero Fortran	348.-
SPC Modula 2	348.-
Metacomco Pascal	298.-
Pro PASCAL	248.-
OS-9	1598.-
UNIX ST (UNIX für ATARI ST)	a. A.

Utilities

Harddisk Utilities (Applic. Syst.)	79.-
Crunch (Opt. Harddiskzugriff, Backup)	79.-
Interlink ST - DFÜ d. Spitzenklasse -	79.-
BTX-Manager	398.-
Turbo ST - der Softwareblitter -	79.-
- mehrere Programme auf Knopfdruck - Desk Assist 4 - Managerprogramm -	189.-
Harddisk Seatry - räumt Ihre Platte auf	139.-
Boot It - auch für MATRIX Großbildschirm -	69.-
- unterschiedlichste Systeminstallationen auf Knopfdruck -	
Revolver ST	79.-
UNISHELL	98.-
NEO Desk	89.-

Emulatoren

PC DITTO V3.64	178.-
PC-Ditto II	a. A.
Aladin V3.0 mit ROMs (unterst. Harddisk + ATARI Laserdr.)	598.-
Spektre 128	a. A.

Spiele

Psion Chess	69.-
Flight II	94.-
Scenery Disks	je 59.-
Jet (F16/F18 Flugsim.)	99.-

Kaufm. Software

BS Handel	498.-
BS Fibu	548.-
RSS PLUS	a. A.

CAD + Grafik

TechnoCAD System	1690.-
CADJA	689.-
CAD Project	279.-
Plotter von A0 – A3 preisgünstig	
Arabesque	
Mega Paint II	329.-
STAD	169.-
Imagic	478.-
Creator	229.-
Dürer	69.-
Degas Elite	69.-

HARDWARE

PROFESSIONAL SCANNER CANON IX12F 3998.-

Technische Daten:

Scannerelement:	CCD Zeilensensor
Scannertyp:	Flachbett mit innenlaufendem Schlitten
Belichtungslampe:	Halogenlampe 50 V / 42.5 W
Zulässige Originale:	Bücher, Blätter, Objekte
Abtastfläche:	216 × 297 mm (DIN A4)
Auflösung:	300 × 300 Punkte/Zoll (300 DPI)
Scanoptionen:	300/200/150/75 DPI
Grauwerte:	32 Graustufen, softwaremäßig erhöhbar
Dichteeinstufung:	3 Stufen
Moduswahl:	binär/dithered
Scangeschwindigkeit:	< 16 Sekunden pro DIN A4 Seite
Abmessungen:	368 (B), 570 (T), 83 (H) [mm]
Gewicht:	114 kg
Spannungsversorgung:	100 – 115 V / 200 – 240 V
Bildformate:	IMG, Postscript, Degas, IF, RGH
Software:	Scansoftware, Retuscheprogramm

Speicherkarten steckbar

auf 1 MB für 260/520 STM	a. A.
auf 2 MB für 520 ST+/1040	a. A.
auf 2,5/4 MB	a. A.

Jede Erweiterung einzeln im Rech-ner getestet! Sehr einfacher Einbau ohne Löten. Gut bebilderte Einbau-anleitung. Vergoldete Mikrosteck-kontakte, dadurch optimale Scho-nung des MMU Sockels. Kein Bild-schirmflimmern, keine zus. Softwa-re, keine zus. Stromversorgung!

ECHTZEITUHR plus 129.-

Keine Software nötig bei Betrieb mit Blitter TOS. Jede Uhr im Rechner getestet und gestellt. Interner Einbau ohne Löten. Dadurch freier ROM Port. Hohe Genauigkeit. Schaltjahr-erkennung, Datum, Uhrzeit.

ICD HARDDISKS

herausgeführter SCSI Port zum An-schluß von Standard PC Peripherie, unterstützt bis zu 8 SCSI Geräte
DMA Eingang, DMA Ausgang
100 % kompatibel zu ATARI Harddisks
sehr leiser Lüfter
alle Harddisks laufen auch mit TURBO DOS
eingebaute ECHTZEIT-UHR
Treiber mit oder ohne Verify
extrem leistungsstarke Software

85 MB form. 2498.-
Harddisks ab 50 MB a. A.
Einbau von 50 MB Harddisks in ATA-RI Mega ST preisgünstig und schnell

STREAMER 2498.-

Übertragungsrate **6,5 MB/Minute!**
Komfortable Software Subsystem aus 50 MB
Harddisk + 155 MB Streamer 3798.-

ST HOST ADAPTER

Anschluß von IBM komp. Harddisks an Ihren ATARI ST unterstützt bis zu 8 SCSI Geräte, DMA Eingang, DMA Ausgang, SCSI Port, integr. ECHTZEIT-UHR. Softwareunterstüt-zung für über 100 versch. Harddisks und Controller, TOP Software,
ST HOST ADAPTER 348.-

FLOPPYLAUFWERKE

3.5'' Einzellaufwerk	289.-
5 1/4'' Einzellaufwerk	398.-

COPROCESSOR 68881 ab 498.-

Für 520/1040/Mega ST **inkl. Softwa-relibraries** für Megamax C, Mark Wil-liams C, DRI C, Lattice C, ProFortran 77, AC Fortran, Modula II, CCD Pas-cal. Einfacher Einbau ohne Löten! INFO anfordern!

VIDEOSOUND BOX 248.-

Ihr ST am Fernseher mit durch-schlagendem Sound! Klangkräftige 3 Wege-Box mit integriertem HF Modulator zum Anschluß aller ATA-RI ST an den Fernseher.

BLITTER Nachrüstplatine	79.-
BLITTER	198.-
TURBO-DIZER	348.-
JUNIOR PROMMER	189.-
EPROMKARTE 64 kB	19.80
Eprombank 128 kB	39.-
Epromdisk 576 kB	139.-
Floppyswitchbox, 3 Lw. am ST	89.-
Floppyweime	39.-
Trakball	98.-
Ersatzteile	a. A.
Floppystecker, Floppybuchsen,	
Monitorstecker, Monitorbuchsen	8.90
Floppykabel	19.90
MMU/GLUE/BLITTERSOCKEL	29.90

COMPUTER-PROJEKTIONS-PANEL

Für alle die etwas präsentieren müssen!
Gestochen scharf, 8 Graustufen, hochmodernes DST LCD, mit IR Fernbedienung! Keine Spezialsoftware nötig.
Geeignet für jeden Overhead Projektor.
Invertier Funktion, Löschfunktion, auch an IBM PS/2, IBM PC/XT/AT mit CGA, EGA, VGA Karte anschließbar!
Auflösung 640 ★ 480 Bildpunkte, nur 3 kg Gewicht! **Preis 3798.-**
Passender Metallkoffer 348.-

Komplettes **DTP System** bestehend aus Computer, Festplatte, Streamer, Monitor (oder Ganzseitenbildschirm) Software und evtl. Scanner a. A.
PRIVATLIQUIDATION für Ärzte incl. Unfallabrechnung a. A. **Komplette EDV Auftragsabwicklung** a. A.
Komplette EDV Buchhaltung a. A. Vorführung nach Absprache! Sonderkonditionen für Wiederverkäufer

---

PLATINEN-LAYOUT

auf dem ATARI ST unter TOS:

MPe II

(MPK PCB Editor II)

Der "MPK Printed Circuit Board Editor II" ist ein integriertes CAD/CAE-Programmpaket für die Leiter-platten-Entwicklung.
In der neuen Version 2.0 bietet es Ihnen:
- Schaltplanerstellung in DIN-Formaten bis A0
- PCB-Design für Leiterplatten bis 508x422 mm
- Feinleiter, SMT, Siebdruck, Lötstop, Bohrdaten

Ausgabe auf 24-Nadel-Matrixdrucker mit stufenlos variablen X- und Y-Skalierungsfaktoren von 0,100 bis 10,000, sowie definierbarer Druck-dichte und Druckqualität.

Komplett für nur DM 723,– !

Voll lauffähiges "EVALUATION-KIT" verfügbar (wird bei Kauf angerechnet).

MPK
Marek Petrik
Vogelsbergstr. 13
3550 Marburg 7
06421/47588

Layoutausschnitt und Hintergrundschaltplan sind unretuschierte Originalausdrucke

---

ATARI ST SOFTWARE

AS-HAUSHALT (Haushaltsbuchführung) **DM 99.-**
Buchführung für alle Privathaushalte. Buchen über Konto/Gegenkonto. Suchen, Löschen und Ändern von Buchungen. Auto-Save. Monats- und Jahresabschluß. Universelle Druckeranpassung. Journalausdruck. Statistik als Balkengrafik. Deutsches Handbuch.

ST-FIRMENBUCH (Buchführung für Gewerbetreibende) **DM 139.-**
Doppelte Buchführung für Gewerbetreibende. Automatisches Mitführen der MwSt.-Konten. Fünf versch. MwSt-Sätze frei einstellbar. Universelle Druckeranpassung. Abschlußzeitraum Monat, Quartal oder Jahr. Einnahmen/Überschußrechnung. Ausgabe eines Journals. Umsatzsteuervoranmeldung. Ausführliches deutsches Handbuch.

ST-ÜBERWEISUNGSDRUCK **DM 45.-**
Bedruckt alle Arten von Überweisungsträgern, Schecks, Zahlkarten usw. Universelle Druckeranpassung, eigene Formulare können mit einem Texteditor selbst erstellt bzw. angepaßt werden. Umwandlung des Betrages in ein Zahlwort. Die Eingaben sind speicherbar.

DATENBANKANWENDUNGEN **DM 69.-**
6 Anwendungen für Adimens ST, z.B. Termine/Adressen, Lager, Videocassetten, Schülernoten, Zeitschriftenartikel, Bibliothek.

ST-SCHREIBMASCHINE **DM 59.-**
Mit diesem Programm arbeitet Ihr Atari ST + Matrixdrucker wie eine Schreibmaschine. Sehr gut geeignet zum Ausfüllen von amtlichen Formularen. Ausdruck entweder direkt oder über editierbares Display. 10 Floskeltasten definierbar. Druckeranpassung. Deutsche Anleitung.

ST-ETIKETT Bedruckt alle Arten von Etiketten. Ausgabebild selbst anpaßbar. Mit Seriennummerngenerierung. **DM 59.-**

***NEU:* ST-TRAINER MATHEMATIK** **DM 79.-**
Lern- und Trainingsprogramm für Schüler (1. bis 5. Schuljahr). Abfrage der vier Grundrechenarten, Bruchrechnen, Kürzen, Erweitern, Längen-, Flächen-, Raum- und Gewichtsmaße (Umrechnungen). Die Abfrage- und Benotungsbereiche sind frei einstellbar. Protokolldruck optional. GEM. Mit deutschem Handbuch.

INFO 1/89 kostenlos!
Lieferung per Vorauskasse (V-Scheck) oder per Nachnahme (plus DM 5.- Versandkosten)!

AS DATENTECHNIK
AS-DATENTECHNIK, MAINZER STR.69, D-6096 RAUNHEIM, Tel.: 06142 / 2 26 77

ATARI ST SOFTWARE

Sie suchen ein leistungsfähiges Programm, mit dem Sie Ihre gesamte Auftragsabwicklung durchführen können???
Dann sollten Sie

ST-AUFTRAG

unbedingt kennenlernen!!!

ST-AUFTRAG ist ein Programm zur Auftragsbearbeitung (Angebote, Auftragsbestätigungen, Lieferscheine, Rechnungen, Bestellungen, Gutschriften) mit integrierter Artikel- und Adressenverwaltung. Der Dialog zwischen dem Anwender und ST-Auftrag wird über die komfortable Benutzeroberfläche GEM geführt. Zusätzlich sind alle wichtigen Funktionen über Tastaturkommandos erreichbar. ST-Auftrag enthält weder Kopierschutz, Hardwareschlüssel noch sonstige Anwenderhindernisse. Die Installation des Systems auf Festplatte in jedem beliebigen Ordner ist problemlos möglich.

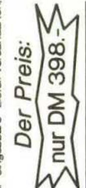
Leistungsbeschreibung in Stichworten:
Max. Anzahl von Artikeln und Adressen: 65535; Artikellisten und -etiketten; Adressenlisten, -etiketten und Serienbrieferstellung; Offene-Posten-Liste; Bestellvorschlagsliste; autom. Mahnwesen; Lagerüberwachung; Rabattliste für jeden Artikel separat; alle Ausgaben auf Drucker, Bildschirm oder Datei; Ausgaben mit 1st Word Plus Formularen (d.h. alle Schriftattribute, Schriftbreiten, Kopf- und Fußzeilen werden mit ausgegeben); Originalgetreue Ausgabe auf Bildschirm; Artikel mit Langtext (ab Version 1.10); Barverkauf; 10 frei eingebbare Liefer/Versandarten; Ausfüllen von Nachnahmezahlkarten...

Hardwarevoraussetzungen:
Monochrommonitor, doppelseitiges Diskettenlaufwerk, Festplatte empfehlenswert, ATARI ST oder MEGA ST mit mindestens 1 MB RAM.

Der Preis:
nur DM 398.-

Und damit Sie ST-AUFTRAG ausgiebig testen können, gibt es eine Demoversion, die keinerlei Funktionsbeschränkungen hat (max. Anzahl Adressen 2, Artikel 5). Die Demoversion kostet DM 20.- (mit Handbuch DM 50.-). Die Preise für Demoversion und Handbuch werden beim Kauf voll angerechnet!

Versand per Vorauskasse oder per Nachnahme (plus DM 5.-).



AS-DATENTECHNIK, MAINZER STR.69, D-6096 RAUNHEIM, Tel.: 06142 / 2 26 77

WARUM

denn immer so kompliziert?

Nehmen Sie doch PegaSoft-Programme !
Die sind schnell und einfach - weil nichts Überflüssiges stört

PegaSoft Rudolf Gärtig Software-Entwickl.
Ringstr. 4 7450 Hechingen 11 07477/8158

PegaSoft

PegaFAKT

Fakturierung mit Lager- u. Adressverwaltung, Etiketten- und Listen-druck, Rechnung mit Netto- oder Bruttopreisen, 3 USt.-Sätze, alle Rechng.-Artikel werden gleichzeitig am Bildschirm angezeigt und kön-nen nachträglich geändert werden, Lieferscheine mit u. ohne Preis, Versandaufkleber m. großer PLZ und Nachnahmebetrag, verbucht auf Knopfdruck (auch Gutschriften), alle PRG-Teile über Funktionstasten erreichbar, Schnittstelle zu PegaSTIC u. Sortierprg. mit Listengener.
Mit Tastatuschablone u. dt. Handbuch mit Schnellkurs nur 99,-

PegaSTIC

Universelles Etikettendruckprogramm, für alle Etikettengrößen bis zu 6 Bahnen u. alle Drucker, bis zu 48 versch. Schriften, automatische Numerierung möglich (beleb. Startwert u. Schrittweite), beleb. Wie-derholungsdruck, eigener Texteditor, Etiketten abspeicherbar, kon-vertieren in andere Größen möglich, vielfältige Gestaltungsmöglichk. (z.B. druckerspezifische Zeichen, Tabellendruck,...), Adressen und Lagerartikel aus PegaFAKT einbindbar (m. Konvertier-PRG f. PD-Vers.)
Diskette mit deutschem Handbuch und Zeilenlineal nur DM 49,-

PRGe werden auf 1-seitiger Diskette geliefert und sind auf JEDEM ST mit Monochrommonitor lauffähig! Sie erhalten die Programme bei Ihrem Händler oder direkt bei uns (Vorkasse 3,-/NN 5,50). DEMO PegaFAKT 20,- (wird bei Kauf angerechnet). Händleranfragen sind erwünscht!

Immer up to date

Mit dieser Sparte wollen wir allen unseren Lesern die Möglichkeit geben, sich über die neuesten Programm-Versionen zu informieren. Angegeben werden die aktuelle Versionsnummer, ein eventueller Kopierschutz, die Bildschirmauflösungen und der Speicherbedarf. Softwarefirmen ist es somit möglich, die ST-Computer-Leser über ihre Updates zu informieren.

Programmname	Version	Daten
Adimens ST	2.3	N HM
Aditalk ST	2.3	N HM
Adress ST / Check ST	1.0	N H
Afusoft Morse-Tutor	2.0	N HML
Afusoft Radio-Writer	1.0	N HML
Afusoft Radiofax plus	1.0	N HML 1M
Aladin	3.0	J H
AnsiTerm	1.4	N
Assembler Tutorials	1.05	N
Banktransfer	1.0	N H
1st BASIC Tool	1.1	N HML
BTX/VTX-Manager	3.0	N H 1M
Calamus	1.09.2	N H 1M
Cashflow	1.0	N H 1M
CIS-L&G	1.01	
Creator	1.0	N H
Crypt_it	1.0	J HML
dBMAN	5.10	N HML
fibuMAN	3.0	N H
fibuSTAT	2.3	N H
Flash-Cache/Flash-Bak	1.0	N HM
Flexdisk	1.3	N HML
FLUSI - Segelsimulator	1.35	N H
1st_Freezer	2.0	N HML 1M
GEMinterface ST	1.1	N HML
GFA-Artist	1.0	N L
GFA-Assembler	1.2	N HML
GFA-BASIC 68881	1.3	N HML
GFA-BASIC-Compiler	2.02	N HML
GFA-BASIC-Interpreter	3.06	N HML
GFA-Draft	2.1	N
GFA-Draft plus	3.0	N
GFA-Farb-Konverter	1.2	N H
GFA-Monochrom-Konverter	1.2	N ML
GFA-Objekt	1.2	N HM
GFA-Starter	1.1	N HML
GFA-Vektor	1.0	N
G+Plus	1.2	N HML
Hänisch Modula-2	N.1	N HML
Hard Disk Accelerator	1.0	N HML
Hard Disk Toolkit	1.05	N HM
Harddisk Utility	2.2	N HM
Imagic	1.1	N HML
Intelligent Spooler	1.10	N HML
Interlink ST	1.89	N HM
K-Resource	2.0	N HM
Kleisterscheibe	2.0	N HM
Label ST	1.0	N HML
Laser C (Megamax)	2.1	N HML
1st_Lektor	1.2	HM
Lern ST	1.22	N HML
Link_it GFA	1.1	N HML
Link_it Omikron	2.0	N HML
Lock_it	1.0	J HML
Mega Paint	2.10	N H 1M
Mega Paint Junior	1.0	N H 1M
Megamax Modula 2	3.5	N HM
Micro C-Shell	2.70	N HM
MT C-Shell	1.20	N HM 1M
Multi ST	1.0	N HML 1M
Multidesk	1.7	N HML
Musix32	1.01	J H
NeoDesk	2.0	N HML
Omikron Assembler	1.05	N HML
Omikron BASIC Compiler	3.04	N HML
Omikron BASIC 68881-Compiler	3.04	N HML
Omikron BASIC Interpreter	3.01	N HML
PAM's TERM/4014	3.012	N H
PAM's TurboDisk	1.7	N HML
PAM's NET	1.0	N HML
PCB-layout	1.17	N H
PC ditto Euroversion	3.96	N HML
PegaFakt	1.3	N H
phs-BTX-Box	6.0	N HML 1M
phs-ST-Box	1.2	N HM
phs-Boxtalk	1.0	N HM 1M
phs-Boxedi	1.0	N HML 1M
phs-Cheapnet	1.2	N HM
Pro Sound Designer	1.2	L
Pro Sprite Designer	1.0	L
Protos	1.1	N H 1M
Revolver	1.1	N HML 1M
Search!	2.0	N HM
Signum! zwei	1.0	N H
Soundmachine ST	1.0	N HM
SPC-Modula-2	1.42	N HML
Spectre 128	1.9	J HM
1st_Speeder	1.01	N HML
1st_Speeder 2	1.0	N HML 1M
STAD	1.3+	N H
Steuer-Tax 2.8	10.2	N HM
Steuer-Tax 3.8	10.2	N HM
STop	1.1	N HM
ST Pascal plus	2.06	N HM
SuperScore	1.4	J H 1M
Tempus	2.05	N HM
TIM	1.2	N H
TIM II	1.0	N H 1M
Transfile ST 1600	1.1	N HM
Transfile ST 850	1.1	N HM
Transfile ST plus	3.0	N HM
Turbo C	1.1	N HM
Turbo ST	1.6	N HML
VSH Manager	2.01	N HML 1M

Irrtum vorbehalten!

Daten-Legende : N = kein Kopierschutz, J = Kopierschutz, H = hohe Auflösung, M = mittlere Auflösung, L = niedrige Auflösung, 1M = mindestens 1 Megabyte

THAT'S Fun-face

Quick Brown Fox © 1989

if He knew of this before, the world surely would look a lot more fun












Presented by COMPO

- GESICHTS MENÜ: 5.576.787.923.200.00 mögliche Gesichter

Läuft auf allen ATARI St/Mega in hoher, mittlerer und niedriger Auflösung (auch auf Stacy) ● Schieben Teile links, rechts, auf, ab, nach innen und außen ● Auswahl per Ziffer oder Zufallsgenerator ● Zufallsanimation ● Sichern und Laden als FAC ● GRAFIK MENÜ: Freihand, Sprühdose & Füllen mit Muster ● Primitives ● Text ● Vergrößern ● Ausschneiden, Kopieren & Einfügen von/nach Clipboard ● Outline ● Bas-Relief ● Spiegeln & Flip-Flop ● Abdunkeln & Aufhellen ● Pixelate & Raster ● Stauchen & Strecken ● Schatten & Multischatten ● 5 Bildschirme ● Sichern/Laden von IMG, PIx, RGH, CLP & ICN

THAT'S ADDRESS

That's Address, die Adressverwaltung zu That's Write

- ☐ Direkte Übernahme einer Adresse von That's Address in den Brief ohne die Daten noch einmal schreiben zu müssen, oder das Programm zu verlassen.
- ☐ Aus diesen Adressen nach eigenen Kriterien bestimmte Einträge für einen Serienbrief selektieren.

Unter Berücksichtigung dieser Vorgaben für That's Write haben wir **That's Address** entwickelt:

Ständig verfügbar
That's Address ist ein **Accessory**, d. h. aus einem laufenden Programm kann auf die Daten zugegriffen werden. Das Programm wird beim Einschalten des Rechners einmal automatisch gestartet und bleibt bis zum Ausschalten verfügbar.

oder als Programm
That's Address läuft auch als Programm, statt dem Accessory kann es (bei zu geringem Speicherplatz z. B.) als Programm gestartet werden.

Hohe Datensicherheit
Jede Änderung oder Neuaufnahme einer Adresse wird sofort abgespeichert.

Einfache Bedienung
wahlweise per Maus oder Tastatur lassen sich alle Funktionen betätigen.

Adressübergabe an That's Write
Wählen Sie den Knopf **Übergabe** – Sie wechseln automatisch zurück in die Textverarbeitung und die angezeigte Adresse erscheint an der gewünschten Stelle in Ihrem Text.

Serienbrief/Datenexport
Ausgewählte Adressen (siehe Selektion) können an That's Write übergeben werden. In Verbindung mit einem von Ihnen gestalteten Text macht That's Write daraus ein Rundschreiben.

Selektion
Adressen können nach freien Kriterien ausgewählt werden. Ausgewählte Listen können sofort bearbeitet werden oder zur späteren oder mehrfachen Verwendung (z. B. erst bearbeiten, dann Etikett, Serienbrief) abgespeichert werden.

Manuelle Selektion
Einzelne Adressen können von Hand (auch mehrfach) z. B. für Etikettendruck abgespeichert werden

Formulardruck
Per Knopfdruck bedruckt That's Address Briefumschläge, Karteikarten, Überweisungen mit beliebigem Absender/Empfänger

Listen-/Etikettendruck
Ausgewählte Adressen können auf den Drucker als
- Endlos-Etiketten
- Liste

ausgegeben werden.

Formularanpassung
Die Formulare wie Überweisung,... können über That's Write von jedem selber geändert werden.

Übersichtlicher Bildschirmaufbau
- Alle Feldbeschriftungen wie Straße, PLZ, Ort,... sind in kleiner Schrift gehalten (man kennt die Felder nach mehrmaliger Benutzung sowieso auswendig) und können vom Benutzer einfach geändert werden.
- Alle Daten wie **Peter Müller, Dorfstr. 34, ...** sind in normaler Schrift gehalten.

Dadurch wird der Bildschirmaufbau nicht überladen und wichtige Daten sind auf einen Blick zu erkennen.

Für den Einsatz von That's Address empfehlen wir einen ATARI ST mit 1 MB RAM (läuft ohne That's Write notfalls auch mit 512 kB)

* alle Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir: _____ *That's Fun Face* á 128,- DM
_____ *That's Address* á 189,- DM

zuzügl. Versandkosten 5,– DM (unabhängig von der bestellen Stückzahl)

Name, Vorname ____________________

Straße, Hausnr. ____________________

PLZ, Ort ____________________

Benutzen Sie auch die in ST-COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51/5 60 57

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

INSERENTENVERZEICHNISS

ST-COMPUTER PUBLIC DOMAIN

Liebe PD-Freunde,

wir freuen uns, Ihnen hier gleich zwei Neuheiten unserer Sammlung zu präsentieren.

Zum einen haben einen wichtigen Entschluß gefaßt, nämlich die ST-Computer PD-Sammlung ab sofort auf doppelseitiges Format umzustellen. Nach langem Abwägen der Laufwerksverteilung haben wir uns trotz knapp 100.000 verkaufter SF 354 zu doppelseitig bespielten Disketten durchgerungen. Die Besitzer von einseitigen Laufwerken mögen dies entschuldigen. Viele PD-Programmierer wird dies indessen sehr freuen, da sie bislang ihre Programme auf etwa 400k beschränken mußten und so manche Grafikeinlage kürzten. Dem ist hiermit abgeholfen.

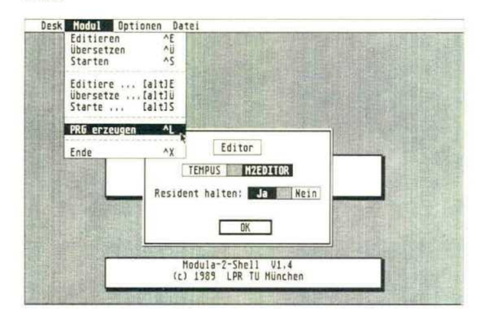
Zum anderen haben wir noch einen anderen Leckerbissen für Sie bereit. Ab sofort liegt jeder PD-Bestellung ein Exemplar der kostenlosen PD-NEWS bei (solange Vorrat reicht). Darin finden Sie auf fast 100 Seiten alles Wissenswerte über unsere Sammlung sowie eine komplette Liste unserer Disketten. Die folgenden PD-NEWS werden vierteljährlich erscheinen.

Viel Spaß



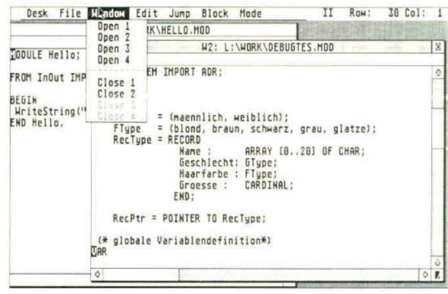
Modula-2-System V 1.4

Bislang auf PD 209 und PD 210 vertreten, finden Sie hier eine stark erweiterte Version des PD-Modula-2-Compilersystems. Der Compiler stammt aus dem Herzen der Modula-2-Sprache, der TH Zürich und wurde von der TU München portiert. Das System beinhaltet alles, was man zum Lernen und auch zum professionellen Arbeiten mit Modula-2 benötigt. Vom speicherresidenten Compiler, dem integrierten Editor und Linker über die umfangreichen Bibliotheken, bis hin zum Source-Level-Debugger, der hilft, auch dem letzten Fehler den Gar auszumachen.



Die Shell des Systems

Neu ist vor allem der integrierte Editor ALTEX, der neben schnellem Scrolling im GEM-Gewand auch sonst hohen Komfort bietet. Außerdem wurden die letzten bekannten Bugs entfernt, so daß man von einer Endversion reden kann. Nähere Informationen finden Sie im Modula-Kurs in ST-Computer 6/89. (1MB)



Der neue Editor kann sich sehen lassen

PD 209/210 sind ab sofort nicht mehr im Programm.



EMACS

EMACS 3.9+: Der EMACS ist der am weitverbreiteste Editor auf allen Rechnersystemen, dessen Features man hier aus Platzgründen nicht beschreiben kann. Auf der Diskette befindet sich die neue Version 3.9+, samt allen Source-Codes und ausführlichster Anleitung. Da die Daten dieser Diskette über 1.3 MByte enthält und wir Ihnen die Source-Codes nicht vorenthalten wollten, wurde Sie gepackt. Mit dem beiliegenden Programm ARC entfaltet sie ihre volle Länge.



EMACS ist wohl der bekannteste Editor, der für fast alle Rechnersysteme existiert.



Utilities

VSCOPE: Bootviruswarner. Das Programm, aus dem Auto-Ordner gestartet, lauert im Hintergrund und prüft bei Diskettenzugriff, ob ein Virus auf dem Bootsektor sitzt. Wenn ja, dann löscht er ihn auf Tastendruck oder installiert ein Schutzprogramm.



Bootviren haben keine Chance mehr

MAUSTREIBER: Ermöglicht die Einstellung der Mausgeschwindigkeit. Die Maus bewegt sich dadurch schneller auf dem Bildschirm. Getrennte Einstellung der X- und Y-Richtung sowie der Schwelle, ab der die Beschleunigung eintritt.

PIC_LINE: Korrigiert automatisch gescannte Bilder, die Zeilenverschiebungen aufweisen. Diese können durch mechanische oder softwaretechnische Probleme entstanden sein. Mit PIC_LINE erhalten Sie wieder ein korrektes Bild. (s/w)

ST-HERCULES: Konvertiert PC-Hercules-Bilder ins ST-Format und zurück. (s/w)

TROBIN: Das Programm konvertiert DRI.o-Format nach .bin-Format.

GETSYSTEM: Zeigt die TOS-Systemversion und die RAM-Größe an.

FREERAM: Klein aber fein. Wie oft kommt es gerade bei der Programmentwicklung vor, daß Sie nicht wissen ob noch genügend RAM zur Verfügung steht. FREERAM zeigt diesen an (als ACC oder PRG).

MAUSPOS: Zeigt die Position der Maus in jedem Programm. Interessantes Programmiertool.



Utilities

DISAS: Disassembler der Spitzenklasse. Darstellung als Programmcode, ASCII, Byte-, Word- oder Long-Konstanten (auch Bit-Notation). Markierung der verschiedenen Datenbereiche. Datei- oder Speicherdisassemblierung, Ausgabeumlenkung, Labelgenerierung und vieles andere. Ein Disassembler, der wohl keine Wünsche offenläßt. Die BASIC-Version findet Verwendung im neuen SED (Scheibenkleister II), und das soll schon was heißen.

PROFILER: Profiliersystem für Pascal-Quellcodes. Untersucht Laufzeitverhalten von Pascal-Programmen. Ideales Hilfsmittel zur Programmoptimierung. Fügt in ein Pascal-Programm Zählbefehle ein, die nach erneuter Compilierung ein Wertefile erzeugt, das schließlich analysiert wird. So kann z.B. eine Tabelle ausgegeben werden, die ausgibt, wie oft verschiedene Schleifen und Prozeduren durchlaufen werden. Ideales Hilfsmittel zur Programmanalyse und -optimierung.

JANINE: Accessory, das ein beliebiges Programm startet. So kann z.B. GFA-BASIC wie ein Accessory gestartet werden.

MABOOT: Bootgenerator, der beim Booten einen Text ausgibt ('Kein Virus im Bootsektor'), ein Bild nachlädt oder ein Paßwort abfragt. Man erreicht damit einen Schutz vor Bootviren oder einfach nur eine originelle Gestaltung des Bootvorgangs.

R_A_U_R: Analyse- und Reassemblierungsprogramm für Resourcefiles. Beliebige RCS-Files können damit untersucht werden und in Assembler-Quellcode verwandelt werden, um in eigenen Programmen Verwendung zu finden.

OC: Object-Cache für Hänisch-Modula. Das Programm puffert für die Object-Files die Ausgabe. Dadurch wird der Compilierungs- und Linkvorgang bei Hänisch-Modula erheblich beschleunigt.



Etiketten

A_MANAGER: Adressverwaltungsprogramm mit Schnittstelle zu Label_Expert 5.0. Sehr schön gestaltetes Programm mit dem sich leicht Adressen verwalten lassen. Der Hintergrund des Programms ist es, das Labeldruckprogramm Label_Expert mit Daten zu versorgen, so daß man sich dort das lästige Tippen spart. (s/w)

LABEL_EXPERT 5.0: Aktuelle, mit A_MANAGER kommunizierende Version des bekannten Labeldruckers. Das Programm bedruckt so ziemlich alle Labels, die man nur bedrucken kann, von Disketten (3.5" oder 5.25") über Videohüllen verschiedenster Bauart, Paketaufkleber, Briefetiketten, bis hin zu Aktenordnern. (s/w)



Kein Etikett entkommt unbedruckt

Machen Sie mit

Möchten Sie ein selbstgeschriebenes Programm in unsere PD-Sammlung geben, um es auch anderen Usern zugänglich zu machen? Kein Problem. Schicken Sie es uns auf einer Diskette zu, samt einer Bestätigung, daß es von Ihnen geschrieben wurde und frei von Rechten Dritter ist. Bei Fragen steht Ihnen die Redaktion gerne zur Verfügung.

ST-COMPUTER PUBLIC DOMAIN



Textverarbeitung

Ihnen wird die Auswahl ähnlich schwer fallen wie uns. Jedes Programm hat seine Qualitäten und Vorzüge, je nach Anwendungsgebiet (wie im käuflichen Bereich).

MINITEXT: Leicht bedienbares Textprogramm mit überzeugenden Leistungsmerkmalen, so z.B. Block-/Flattersatz, verschiedene Schriftarten, Laden von Wordplus-Texten, Aufruf eines anderen Programms, Blockoperationen, Textbausteine und umfangreiche Druckeranpassung. Nicht zuletzt ist ein UPN-Rechner integriert. (s/w)



Der Rechner leistet oft wertvolle Hilfe

PD-TEXT 2: Textverarbeitung mit integriertem Karteikasten. Das Programm beherrscht alle Schriftarten auf dem Bildschirm, Blocksatz, Blockoperationen, Suchen/Ersetzen usw. Auch ein Taschenrechner fehlt nicht. (s/w)

--> Bild siehe nächste Spalte



PD-Text wird interpretiert, ist aber dennoch schneller als manch anderes Programm

PROFITEXT: Programm mit besonderen Leistungsmerkmalen. Neben den normalen Textfunktionen beherrscht Profitext ein Wörterbuch mit automatischer Korrektur, Trennhilfe, Flosкeln, große Headlines beim Druck, Zeichensatzeinbindung sowie Spaltensatz. Eine frei definierbare Druckeranpassung macht dies bei allen Druckern möglich. (s/w)



Profitext trägt seinen Namen zu Recht.



Adressverwaltung

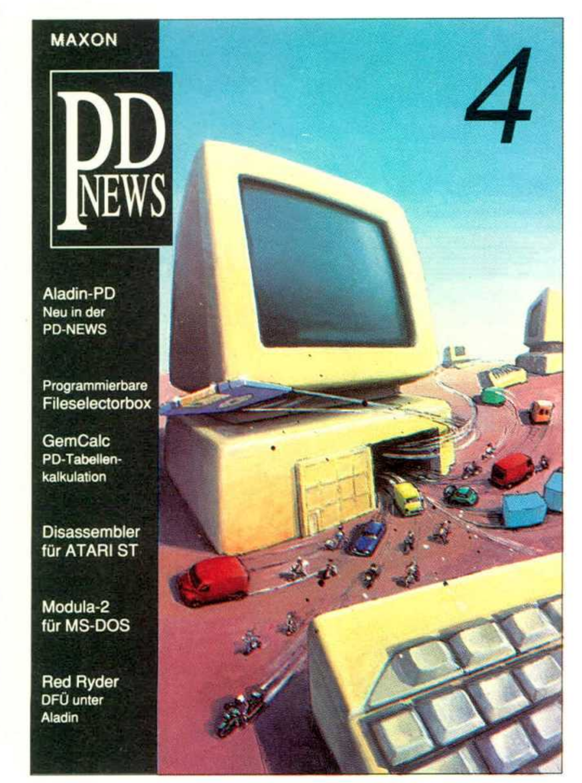
ADRESSBÜCHLE: Sehr schöne Adressverwaltung. Aufbau wie die bekannten Dateiboxen, die meist an Telefonen zu finden sind. Nach Klick auf den Buchstaben öffnet sich die Adressmaske, die sich per Pop-Up-Menüs leicht blättern läßt. Source-Code in GFA-BASIC 2.0 und 3.0 beiliegend. (s/w)



Adressbüchle ist praktisch zu bedienen

ADRESSDATEI: Adressverwaltung mit verschlüsselten Daten. Somit sind alle Daten vor unbefugten Gebrauch gesichert. Etikettendruck. (s/w) (1 MB)

Zeichenerklärung:
s/w = nur monochrom
f = nur Farbe
1MB = mind. 1MB freier Speicher nötig



Die PD-NEWS wird bei jeder Bestellung (solange Vorrat reicht) **kostenlos** mitgeliefert!

PD-NEWS 4, jetzt kostenlos!

Tja, endlich ist sie da, die neue PD-NEWS 4. Sie enthält auf fast 100 Seiten viele Berichte, Tips und Tricks zu PD-Programmen und - nicht zu vergessen - die komplette Liste der ST Computer-PD-Sammlung. Viel hat sich geändert seit der letzten Ausgabe: Das Wichtigste ist mit Sicherheit, daß die PD-NEWS jetzt kostenlos bei Ihrer Bestellung beigelegt wird. So sind Sie immer auf dem neusten Stand, was PD angeht. Im folgenden wollen wir Ihnen einen kurzen Einblick auf den Inhalt geben:

- Neue PD-Sammlung für Aladin-Anwender (31 Disketten zum Beginn, wird ständig erweitert)
- Red Ryder - DFÜ unter Aladin

- Programmierbare Fileselectorbox
- GemCalc - die kostengünstige Tabellenkalkulation für den ST
- PD-Disassembler - den Programmen ins Listing geguckt

- PCD - Software für MS-DOS-Emulatoren
- FST-Modula-2 - nun auch ein PC-Modula als PD

- Komplette Liste der Kickstart-PD-Sammlung für den AMIGA

u.v.m.

Auf ca. 100 Seiten das Neueste aus der PD!

ST-COMPUTER PUBLIC DOMAIN

DATEBOOK: Das Programm meldet beim Systemstart automatisch alle Termine der nächsten Tage, egal ob Geburtstage oder sonstige wichtige Ereignisse. Dabei lassen sich auch immer wiederkehrende Termine (z.B. Feierabend) eingeben. Sehr nützlich. (s/w)

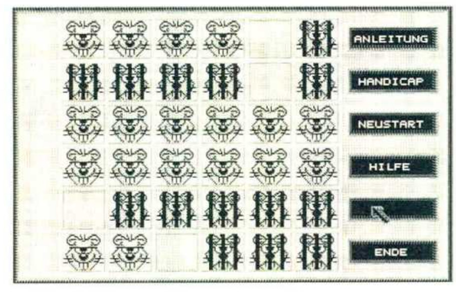


Datebook erinnert an wichtige Termine



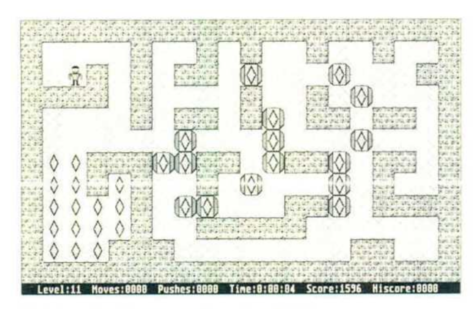
Spiele

RATTE: Sehr interessantes Kombinationsspiel. Die Rattenplage herrscht im Land. Sie haben die Aufgabe durch gezielte Bewegung der Spielsteine, die Ratten hinter schloß und Riegel zu bringen. Doch das geht nicht immer so leicht wie man denkt. Faszinierende Spielidee. (f)



Ratte fördert die Gehirndurchblutung

THINKWORK: Aufgabe ist es, in einem Labyrinth Steine durch Verschieben an die Zielpositionen zu bringen. Dazu gehört allerdings einiges an Gehirnschmalz, da man sich schnell den Weg verbauen kann. Joystick- oder Tastatursteuerung. 25 Level, weitere können mit dem eingebauten Leveleditor erzeugt werden. Der Highscore jedes Levels wird abgespeichert.



Thinkwork- ein anspruchsvolles Spiel

ALEA: Zauberwürfelähnliche Gehirnauflockerung. (s/w)

GEGEN 4: Tron-Variante für 4 Spieler. Das Spiel bekommt neue Dimensionen, da Sie gegen andere Spieler plus den Computer spielen müssen.

NIM: Altbekanntes Streichholzspiel, bei dem Sie immer etwas schlauer als der Computer sein müssen. (s/w)

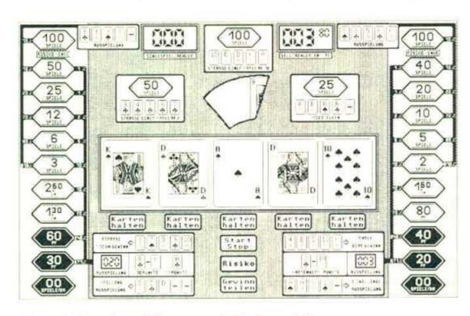
Eine komplette Übersicht über unsere PD-Sammlung (1-236 finden Sie in der PD-NEWS 4. (siehe vorherige Seite)



Glücksspiele

BIBOS CASINO: Beinhaltet drei Glücksspiele: Venus-Multi und Roulett-Multi, zwei Automaten die aus jeder Spielothek bekannt sein dürften sowie Bibos-Slot-Machine, ein einarmiger Bandit mit besonderen Gewinnchancen. (s/w)

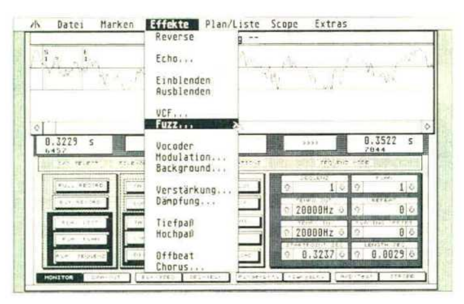
GRAND_H: Spielautomat mit vielen Details. Naturgetreue Simulation des Spielablaufs und der optischen Effekte. (s/w)



Grand Hand verführt zum Glücksspiel

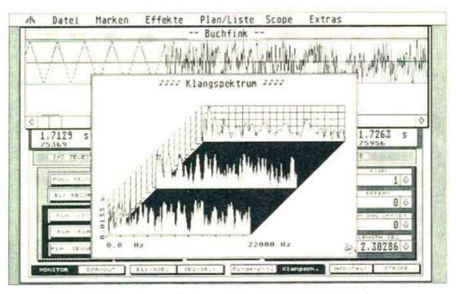


Sampler



SAM bietet eine Vielzahl von Manipulationsmöglichkeiten

SAM: Soundsampler mit überragenden Funktionen, die man eigentlich auf einem 2seitigen Testbericht beschreiben müßte, na vielleicht in der nächsten PD-NEWS. SAM verfügt über exquisite Funktionen beim Samplen z.B. Aussteuerung, Oszilloskop mit manueller und automatischer Triggerung bis hin zum Spekrumanalyzer mit Echtzeit-Fourieranalyse. Auch bei der nachträglichen Bearbeitung der Samples überzeugt SAM durch vielfältigste Möglichkeiten: Echo, Ein- und Ausblendung sind da das wenigste. SAM kann Frequenzen aufmodulieren, Zweitstimmen frequenzverschoben addieren bis hin zum Erzeugen von Rechteck-, Sinus-, Sägezahnkurven und Rauschen. Die Krönung des Ganzen bieten dann der eingebaute Spektrumanalyser, der die Sampledaten 3-dimensional aufzeigt. Das Programm unterstützt sämtliche bekannten Sampler. (s/w)



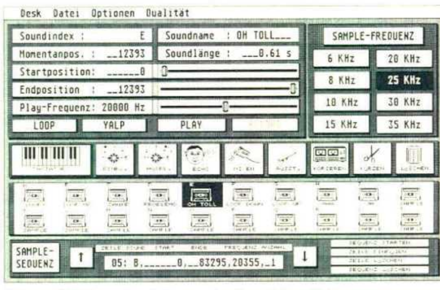
Das Klangspektrum gibt interessante Informationen über das Sample

JINGLEBOX: Mit diesem Programm können die erzeugten SAM-Sounds in verschiedenen Tonhöhen per Tastatur oder Midi-Keyboard abgespielt werden. Aus einem gesampelten Ton läßt sich damit eine Tonleiter bzw. ein ganzes Instrument erzeugen. (s/w)



Sampler

SAMPLE: Ebenso ein leistungsstarkes Programm, das sich besonders zum Zusammenfügen verschiedener Samples zu einem langen Musikstück eignet. Dazu existiert eine Ablaufliste, die Länge und Tonhöhe bestimmt. Der Speicherbedarf nimmt mit Länge des erzeugten Musikstücks nicht zu, daher können minutenlange Samples abgespielt werden. Natürlich verfügt das Programm auch über Funktionen zum Ein- und Ausblenden, Mischen und Kopieren von Sequenzen. Die Bedienung ist bildschirmorientiert und daher unkompliziert. Ein beigefügtes ALF-Sample verschafft einen Eindruck von den Möglichkeites des Programms.



Bis zu 20 Samples können in der Ablaufliste berücksichtigt werden.

BITTE BEACHTEN

Sämtliche Disketten können ab dem Erstverkaufstag der ST-Computer direkt bei der MAXON-Computer bezogen werden.

Bitte beachten Sie folgende Punkte:

1. Schriftliche Bestellung

- Der Unkostenbeitrag für eine Diskette beträgt DM 10,-
- Hinzu kommen Versandkosten von DM 5,- (Ausland DM 10.-)
- Bezahlung nur per Scheck oder Nachnahme
- (Im Ausland **nur** Vorrauskasse möglich)
- **Ab 5 Disketten entfallen die Versandkosten (DM 5.- bzw. DM 10.-)**
- Bei Nachnahme zuzüglich DM 4,00 Nachnahmegebühr
- Jeder Bestellung liegt (solange Vorrat reicht) die neuste PD-NEWS bei.

Bitte fügen Sie keine anderen Bestellungen oder Anfragen bei!

Adresse:

MAXON-Computer GmbH
"PD ST-Computer"
Postfach 5969
D-6236 Eschborn

2. Anruf genügt

MAXON-Computer GmbH
"PD-Versand"
Tel.: 0 61 96 / 48 18 11
Mo-Fr 9⁰⁰ - 13⁰⁰ und 14⁰⁰ -17⁰⁰ Uhr

- Nur gegen Nachnahme (Gebühr DM 4,00)

Listen der Aladin- und PCD- (MS-DOS) Disketten in der PD-NEWS oder gegen ausreichend frankierten Rückumschlag erhältlich.



Sonderdisketten

Was ist besonderes an den Sonderdisketten. Na ja, sie sind etwas ganz anderes. Sie sind unter vielen Einsendungen ausgesucht und für Sie zusammengestellt worden. Sonderdisks kosten wenig aber leisten viel. Es ist im Prinzip eine besondere Ehre aufgenommen zu werden, denn es schafft nicht jeder. Wir laden Sie zu einem Rundblick ein:

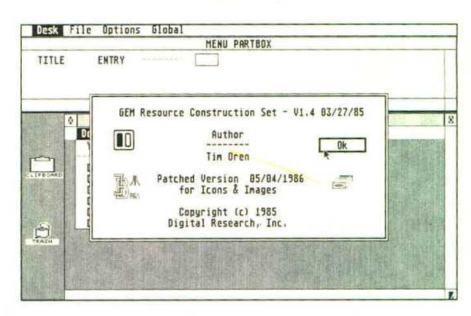
TOS

TOS heißt das Betriebssystem des ATARI ST und ist somit die Software, die ihn zum Leben erweckt. Mit der Zeit haben sich im Betriebssystem Änderungen ergeben, und es kann vorkommen, daß ältere Programme nicht mehr mit dem aktuellen Betriebssystem funktionieren. Was tun? Einfach die TOS-Diskette (Version vom 6.2.1986) einlegen und den Rechner starten und schon ist alles beim altem.



TOS DM 15,-

RCS



Unter der geheimnisvollen Abkürzung RCS versteckt sich das Resource Contruction Set in der Version 1.4 aus dem ATARI-Entwicklungspaket. Mit diesem sinnvollen Utility können Sie ganz einfach Menüleisten und Dialogboxen für eigene Programme erstellen, genauso wie Sie sie von vielen GEM-Programmen her kennen. Das ist ein muß für jeden Programmierer, der seine Software professionell gestalten möchte. Eine ausführliche Beschreibung des Funktionen dieses Programmes finden Sie im Sonderheft Nr. 2 der ST Computer.



RCS 1.4 DM 15,-

Extended VT 52-Emulator

Schneller als der Blitter

Erweiterte Version des in der ST-Computer ausführlich vorgestellten Emulators. Das Programm ersetzt den im TOS integrierten VT52-Emulator vollkommen. Enthält neue Routinen zur Bildschirmausgabe, die wesentlich schneller sind als die im TOS eingebauten. Daher erfolgt eine Beschleunigung der Textausgabe um den Faktor 3 bis 5 je nach Komplexität.

Programmierung

Neben der Beschleunigung verfügt der Emulator über zusätzliche, programmierbare ESC-Funktionen, wie z.B. Softscrolling (1 Pixel) nach allen vier Seiten. Wird es eingeschaltet, scrollt eine Zeile oder ein Textblock ohne ruckartige Effekte. Ferner können Textattribute (fett, unterstrichen, hell und kursiv) ohne GEM umgeschaltet werden. Auch die Übertragung von Grafik via VT52 ist möglich. Der Extended VT52 Emulator wurde in vier Teilen ausführlich in der ST-Computer vorgestellt.



VT52-Emulator DM 15,-

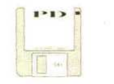
Accessories
Kleine, große Helfer I

Lovely Helper
Rechner | Directory
Info | Spooler
Kalender | Zeit
Abbruch

LOVELY HELPER

Sie kennen das Problem: Sie haben viele verschiedene Accessories, und alle können Sie gebrauchen, und am liebsten möchten Sie alle bei Starten des Rechners mitbooten. Leider mach das Betriebssystem nicht mit, und wie Sie schon wissen, stehen nur 6 freie Plätze für Accessories in der Menüleiste zur Verfügung. Dann muß man das ein oder andere Accessory opfern. Eine Lösung für solch ein Problem heißt Multi-Accessory, und Lovely Helper ist eins davon.

Lovely Helper ist Accessory, daß quasi sechs kleine, große Helfer umfaßt: Einen Taschenrechner, mit dem Sie auf die Schnelle mal 'was ausrechnen können, einen Kalender, den man auf jedes Bundesland abstimmen kann, eine Uhr, die in die Menüleiste eingeblendet wird, eine Directory-Anzeige für jedes beliebige angemeldete Laufwerk, eine Directory-Druckoption, die nach beliebigem Kriterium den Inhalt einer Diskette ausdruckt, und einen komfortablen Druckerspooler. Und wem das alles noch nicht genug ist, der hat dann immer noch Platz für weitere fünf Accessories. Der Quellcode liegt in Pascal bei.



Accessories I
Lovely Helper DM 15,-

Accessories
Kleine, große Helfer II

Das ist unsere zweite Sonderdisk mit Accessories, diesen unentbehrlichen Helfern. Diesmal geht es fast nur um allgemeine Grafik aber auch ums Drucken:

PARTCOPY

Wie Sie wahrscheinlich wissen, erlaubt die eingebaute Hardcopy-Routine des ST das Ausdrucken von ganzen Bildschirmen. Wenn es aber darum geht, nur Teile des Bildschirms auszudrucken, dann ist diese Routine am Ende. PartCopy erlaubt Ihnen, einen bestimmte Ausschnitt des Bildschirms auf den Drucker zu geben oder auf Diskette zu speichern. Es bleibt Ihnen überlassen, wie der Ausschnitt aussehen soll. Zum Ausschneiden stehen Ihnen verschiedene Möglichkeiten zur Auswahl: rechteckig, polygonal und mit einem Lasso. Die erzeugte Bilder können Sie später beliebig verwenden (z.B. in Word Plus einbinden). Der Quellcode liegt in Assembler bei.



SPOOLER

Sie kennen schon das Problem: Sie möchten einen ziemlich langen Text auf Ihren Drucker ausgeben, möchten aber sinnvollerweise gleichzeitig weiter mit Ihrem Rechner arbeiten. Leider geht das in der Regel nicht, da das Betriebssystem diese Möglichkeit nicht vorgesehen hat. Hier hilft ein schneller Spooler. Der vorgestellte Spooler kann nicht nur Texte sondern auch Hardcopies "spoolen". Abarbeitungsgeschwindigkeit und Größe des reservierten Speicherplatzes können Sie selber einstellen. So können Sie immer kontrollieren wie lange die Spooler braucht, um fertig zu werden. Der Quellcode liegt in Assembler bei.

PRINTTOP

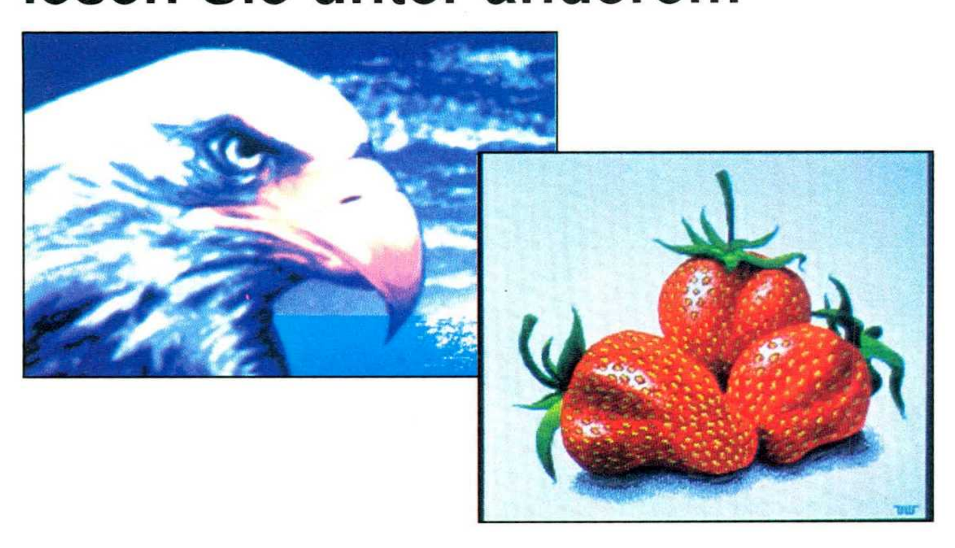
Möchten Sie eine Hardcopy von nur einem bestimmten Fenster? Vielleicht trifft das bei Ihnen öfters zu. Mit diesem Accessory werden Sie immer in der Lage sein, das gerade aktive Fenster auf einen Drucker auszugeben. Der Quellcode liegt in Modula-2 bei.



Accessories II
Partcopy, Spooler & Printop DM 15,-

*Sonderdisks kosten je DM 15,- und sind **nicht public domain**. Die Versandbedingungen entsprechen denen der PD-Disketten, mit Ausnahme der DM 15,-.*

In der nächsten ST-Computer lesen Sie unter anderem



MGE - Die MAXON Grafik Expansion

Bekommen Sie auch Appetit, wenn Sie das Bild sehen. Aber wir wollen Sie nicht zum Essen verführen sondern eine neue Grafikkarte für alle Mega STs vorstellen. Den Prototyp konnte man schon auf der letzten CeBIT bewundern, jetzt ist sie fertig. Mit einer maximalen Auflösung von 1280x1024 Pixeln ist sie für alle Grafikfans interessant, die mehr aus ihrem Rechner holen wollen.

Easy Rider...

... ist kein Programm für Motorradfreunde sondern stellt sich als intelligenter, kombinierter Re- und Disassembler dar. Es vermag selbständig Daten- und Codebereiche von ablauffähigen Programmen zu unterscheiden und eignet sich vor allem zur Optimierung von Programmen, die in höheren Programmiersprachen geschrieben wurden.

Arabesque

"Schon wieder ein Malprogramm!", werden viele sagen. Doch Arabesque arbeitet im Gegensatz zu anderen Programmen dieser Art sowohl mit Vektor- als auch mit Pixelgrafik, wodurch sich ungeahnte Möglichkeiten eröffnen. Mehr darüber lesen Sie nach unserer Sommerpause.

Die nächste ST Computer erscheint nach der Sommerpause am Fr., den 25.08.89

Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen, kostet sehr viel Zeit und Mühe. Da wir ja weiterhin vorhaben, die Qualität zu steigern (ja, auch wenn das manchmal danebengeht), haben wir Redakteure ein großes Anliegen an Sie, liebe Leserinnen und Leser:

Bitte haben Sie Verständnis dafür, daß Fragen an die Redaktion nur Donnerstags von 14⁰⁰-17⁰⁰ Uhr telefonisch beantwortet werden können.

Vielen Dank für Ihr Verständnis

Impressum ST Computer

Chefredakteur: Uwe Bärtels (UB)

Redaktion:
Uwe Bärtels (UB)
Harald Egel (HE)
Marcelo Merino (MM)
Harald Schneider (HS)

Redaktionelle Mitarbeiter:
C.Borgmeier (CBO), Fernando Brand (FB), Claus Brod (CB), Ingo Brümmer (IB), Stefan Höhn (SH), Raymund Hofmann (RH), Oliver Joppich (OJO), Jürgen Leonhard (JL), Claus P. Lippert (CPL), Markus Nerding (MN), Chr. Schormann (CS), R.Tolksdorf (RT)

Autoren dieser Ausgabe:
K.Bärtels, D.Brockhaus, Calvi, B.Dettlaff, Dr.A.Ebeling, S.Geier, O.Glöckler, A.Günthner, A.Hill, M.Kofler, J.Krill, A.Lötscher, Th.Mahler, K.Opel, M.Pfahl, K.Rindfrey, C.Schmitz-Moormann (CSM), U.Seimet, J.Stessun, I.Wedekind

Auslandskorrespondenz:
C.P.Lippert (Leitung), D.Dela Fuente (UK)
L.Hennelly (Nordamerika)

Redaktion: "MAXON" Computer GmbH
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 11, FAX : 0 61 96/4 11 37

Verlag: Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57, FAX : 0 61 51/5 56 89 + 5 60 59

Verlagsleitung:
H.J.Heim

Anzeigenverkaufsleitung:
U.Heim

Anzeigenverkauf:
K.Margaritis

Anzeigenpreise:
nach Preisliste Nr.3, gültig ab 1.1.88
ISSN 0932-0385

Grafische Gestaltung:
Gabriele Imhof
Kerstin Feist, Bernd Weber

Titelgestaltung:
Gunter Wenzel (Tel.: 06172/37193)

Fotografie:
Archiv, dpa

Produktion:
K.H.Hoffmann,

Druck:
Ferling Druck W.Frotscher GmbH

Lektorat:
V.Pfeiffer

Bezugsmöglichkeiten:
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

ST Computer erscheint 11 x im Jahr
Einzelpreis: DM 7,-, ÖS 56,-, SFr 7,-
Jahresabonnement: DM 70,-
Europ. Ausland : DM 90,- Luftpost : DM 120,-
In den Preisen sind die gesetzliche MWSt. und die Zustellgebühren enthalten.

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern der MAXON Computer GmbH. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

DISK HELP

Repariert Disketten und erlaubt in 99% der Fälle eine Wiederbelebung des gesamten Datenbestandes. Das Programm gehört neben jeden ST. Lassen Sie Ihre wichtigsten Daten nicht einfach von Laufwerken zerstören. DISK HELP ist einfach zu bedienen, korrigiert Lesefehler und rekonstruiert Files. Physikalische Fehler (Risse, Kaffee) können nicht behoben werden.

DM 79,—*

FAST SPEEDER

Für alle, die ihrem Massenspeicher keine Pause gönnen wollen und lange Wartezeiten satt haben. Bringen Sie Ihre Festplatte und Diskettenstation auf Trab! Im Extremfall erreichen Sie dabei Ramdiskgeschwindigkeit! Festplatten werden bis zu 200% beschleunigt. Diskettenlaufwerke bis zu 900%. FAST SPEEDER ist einfach zu handhaben. Der optimierte Spezialalgorithmus mit Cache Prinzip arbeitet für Sie unmerklich im Hintergrund.

DM 129,—*

SYMPATIC PAINT

neue Version!

Der ATARI ST als Grafik Workstation der Superlative! Alle Funktionen, die für ein effektives und komfortables Zeichnen, Malen und Konstruieren erforderlich sind, wurden integriert. Der Benutzer hat nicht nur die Möglichkeit der Erstellung einer individuellen Grafiktoolbox, sondern auch eine Animationsselektion, mit der Bilder laufen lernen. Bilder können dabei mit dem G SCANNER und EASYTIZER eingelesen werden.

Neue Version mit über 40 zusätzlichen Funktionen!

Preissenkung: DM 199,—*

Handbuch vorab: **DM 30,—***

G COPY II

neue Version!

Das bewährte vielseitige Kopierprogramm wurde mit einer neuen, stark verbesserten Oberfläche versehen, und mit zahlreichen Verbesserungen sowie neuen Features auf den Level von 1989 gebracht.

Mit seinen zahlreichen Zusatzplänen kann G COPY II nun noch mehr: kopiert und formatiert alle Formate (912 K-, 862 K-, 416 K-Disketten mit 10 oder 11 Sektoren und 80 — 83 Tracks) — optimale Kopiergeschwindigkeit bei 100% Datensicherheit — alle Programme ohne Hardwarekopierschutz werden kopiert. Neu: bis Track 90.

G COPY dient nur zur Erstellung von erlaubten Sicherheitskopien.

Raubkopien sind strafbar!

Für alle ST mit Monochrom- und Farbmonitor.

DM 99,—*

ANTI VIREN KIT II

Das Anti Viren Kit II ist inzwischen zum Standard geworden, und gehört neben jeden ATARI ST.

Das Anti Viren Kit II erkennt und vernichtet alle bekannten Computerviren und deren Verwandte, Datenrestbestände werden gerettet; Linkviren werden abgeschnitten und vernichtet; immunisiert gegen Bootsektorviren; intelligente Bootsektorscannung; auch für Viren auf der Harddisk! Alle registrierten Benutzer können unseren kostenlosen Updateservice nutzen. **Neu: Jetzt mit Vektorüberwachung und Diskettendurchleuchtung.**

Fachpressestimmen: *„Das wohl mit Abstand erfolgreichste Programm zur Bekämpfung von Viren", „...erkennt alle bekannten Viren und vernichtet sie.", „G DATA hat das anerkannt beste und umfassendste Programm gegen Viren."*

DM 99,—*

HANDY SCANNER

Die bekannten, extrem hochwertigen, komfortablen Scanner mit Software. Kompromißlos in Qualität und Handhabung!

Der HANDY SCANNER wird einfach über die abzurasternde Fläche gerollt, und schon erscheint die Vorlage auf dem Bildschirm. Ideal für DTP, zum Scannen von Fotos, Zeichnungen, etc.

Die abgespeicherten Bilddateiern sind übertragbar. Das Grafikpaket HANDY PAINTER ist im Lieferumfang enthalten.

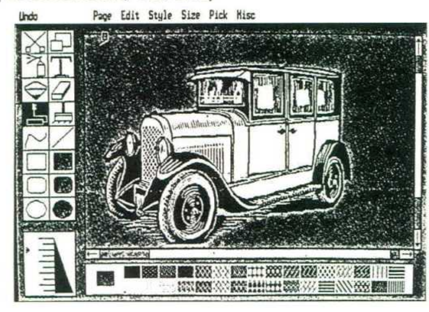
Leistungsdaten:

- Einfacher Romportanschluß
- Kontrast und Empfindlichkeit einstellbar
- Auflösung bei 200 DPI vertikal und horizontal je 8 Punkte/mm, bei 400 DPI jeweils 16 Punkte pro mm Scannfläche
- Einsatz hochwertiger Spezialdioden für maximale Qualität
- 64 mm Scannbreite **(breitere Vorlagen können durch mehrbahniges Einscannen am Bildschirm auf einfache Weise zusammengesetzt werden).**

HANDY SCANNER II (200 DPI, s/w Gerät) DM 598,—*

HANDY SCANNER IV DM 898,—*

(16 Graustufen, 400 DPI!)



Hardware Uhr

Die Uhr wird ohne Lötarbeiten im Rechner eingebaut, so daß der ROM Port frei bleibt. Sie haben ständig die aktuelle Zeit und das aktuelle Datum! Die Batterie garantiert hohe Ganggenauigkeit und mindestens 3 Jahre Laufzeit bei voller Schaltjahrerkennung. Der sensationelle Preis für die anschlußfertige Uhr beträgt nur

DM 79,—*

INTERPRINT II

Die universelle Anpassung für jeden Drucker hat einen integrierten Treiber, der auch alle Umlaute und Sonderzeichen nach Ihren Wünschen konvertiert. Dabei können mit Hilfe des Editors maßstabsgerechte Hardcopies auf allen Matrixdruckern und HP-Lasern erstellt werden. Auch ein Druckerspooler und unsere G RAMDISK II+ sind integriert. Die Parameter sind speicherbar, und der komfortable Editor ermöglicht schnellste Anpassung.

DM 99,—*

ohne Ramdisk **DM 49,—***

RELAS

Aktienanalyse für den privaten und professionellen Aktienanleger auf der Basis der Relativen Stärke und Volatilität von Charts wahlweise über Bildschirm und Drucker – Mischen von Charts einer Aktie und Branche zum Vergleich – Kurseingabe nur einmal wöchentlich (!) Als Basis dienen die letzten 27 Wochenschlußkurse – Umfangreiche Bereinigungsfunktion bei Kapitalveränderungen von Aktien - Listen der Relativen Stärke mit Mittelwert von Kursen und Analyse der Veränderung der Rangfolge der Relativen Stärke über 3 Wochen (wahlweise Bildschirm oder Drucker) – u.v.m.

DM 398,—*

Handbuch vorab: **DM 40,—***

Roboterarm

Der semiprofessionelle 5achsige Roboterarm wird anschlußfertig für ST, AMIGA und PC/AT geliefert. Für Schulungszwecke, Lehr- und Demonstrationszwecke oder einfach zum Spielen bestens geeignet. Technische Daten: Höhe ca. 54 cm, industriegelb, ca. 2,5 kg schwer, einfacher Anschluß an Druckerport. Auch über Joysticks zu steuern.

Anschlußfertig: **DM 269,—***

Spezialnetzteil: **DM 99,—***

Batteriesatz: **DM 10,—***

TV Modulatoren

Schließen Sie Ihren ST an den Farbfernseher an und sparen Sie sich einen teuren RGB Monitor. Unsere Modulatoren bringen ein scharfes Bild mit allen Mischtönen und einen Tonausgang extern und einen über den Fernseher. Das heißt: bestes Bild und bester Ton für alle Fernseher.

Typ A (ATARI an TV) mit Monitorumschalter **DM 225,—***

Typ C FBAS Wandler für Geräte mit AV oder FBAS Eingang **DM 165,—***

AS SOUND SAMPLER

jetzt lieferbar: AS SOUND SAMPLER III Sampling in CD-Qualität

Schon der Sound Sampler II überzeugte Kunden wie Presse (Keyboards 12/87: „Verglichen mit 10-15-fach teureren Samplern ... muß man von einer guten bis sehr guten Qualität sprechen." ST 1/88: „Der Alleskönner!") Alle Leistungsmerkmale des Sampler II bietet der neue Sound Sampler III in 16 BIT CD Qualität! Aufnehmen, editieren, verknüpfen, mischen, transponieren, Echoeffekte, einbinden in Programme, Sounds oder MIDI Keyboards spielen; alles in original CD Qualität! Der AS SOUND SAMPLER II jetzt mit stark verbesserter HiFi Qualität!

AS SOUND SAMPLER II maxi:	**DM 298,—***
AS SOUND SAMPLER III 16 BIT:	**DM 598,—***
Soundbibliothek 10 Disketten für Sampler II	**DM 149,—***
Soundbibliothek 10 Disketten 16 BIT	**DM 198,—***

GEM Retrace Recorder

Der Knüller! Wie von Geisterhand bewegt sich die Maus, laufen Kommandos ab, werden Aktionen gestartet. Der GEM Retrace Recorder ermöglicht es, jegliche Aktion (Mausbewegung, Menüs, Klicks, Tastatur usw.) aufzuzeichnen, und auf Tastendruck jederzeit original zu wiederholen. Das heißt: GEM Makros erstellen für Aktionen, die ständig wiederholt werden (auch in jedem Programm); super Auto Boots erstellen; selbstlaufende Demos von jedem Programm ohne Programmeingriffe und Kenntnisse u.v.m. Die Anwendungsmöglichkeiten sind fast unbegrenzt!

DM 99,—*

Harddisk Help & extension

Wir wollen keinesfalls die Ängste um Ihre täglichen Daten schüren, aber sind Ihre Daten wirklich sicher? Wie schnell führt ein Headcrash zum Ende! Sie sollten auf ein bewährtes Backupsystem nicht länger verzichten. Auch sehr große Files können einzeln gesichert werden. Partition Backup, Tree Funktion, selected Backup und Diskoptimizer erhöhen die Effizienz dieses Programms. Auch bei der Geschwindigkeit wurden keine Kompromisse geschlossen.

Für ATARI SH 205 DM 129,—*

G RAMDISK II+

In der neuesten Version mit allen Leistungsmerkmalen, die man sich für eine Ramdisk überhaupt wünschen kann: Resetfest bis 4 MB, dabei abschaltbar; integrierter Druckerspooler, Bootcopy; Parameter save usw. Die G RAMDISK II ist im harten Einsatz bereits tausendfach bewährt.

DM 49,—*

Coupon

Hiermit bestelle ich

____

❑ Verrechnungsscheck liegt bei (+ DM 5,— Versandkosten)

❑ Informationsmaterial (kostenlos)

❑ 2 Demodisketten und Infomaterial (DM 10,— beiliegend)

❑ per Nachnahme (ca. DM 6,50 Versandkosten)

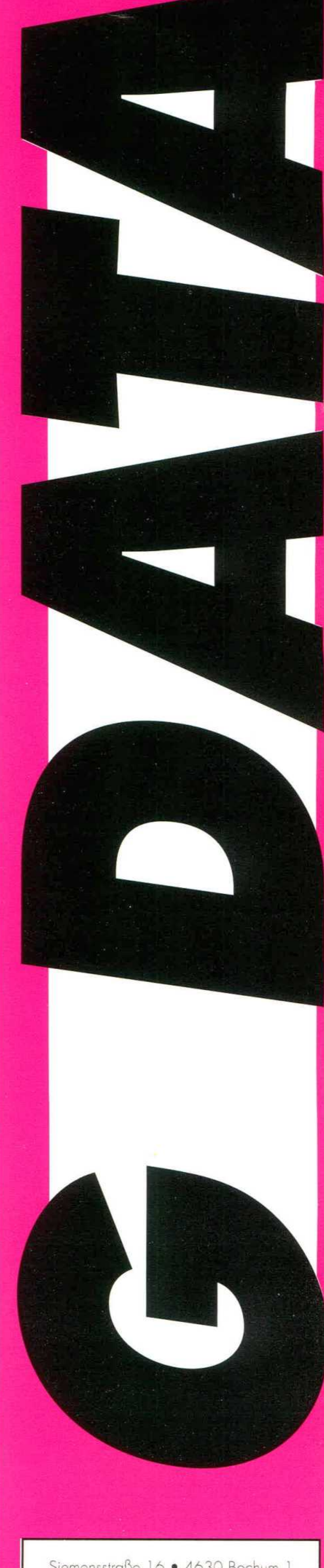
Absender:

____

____

____

*unverbindliche Preisempfehlung
Irrtum und Änderung vorbehalten



Siemensstraße 16 • 4630 Bochum 1

Telefonische Bestellungen:

Tel. 0 23 23 / 38 98 58

Schweiz: DATA Trade Langstraße 94 CH-8021 Zürich

Österreich: Computershop Rittner Hauptstraße 34 A-7000 Eisenstadt

Bei uns gehört der Compiler dazu

GFA BASIC 2.0

Für Einsteiger
Das GFA-BASIC ST Entwicklungssystem 2.0 Interpreter + Compiler DM 49,90 (Upgrade-Möglichkeit zum GFA-BASIC ST Entwicklungssystem 3.0 DM 160,–)

GFA-BASIC, das Standard-Programm, weltweit über 100.000 mal im Einsatz, wurde Software des Jahres 1988 in der amerikanischen Computer-Zeitschrift ANTIC.

GFA BASIC 3.0

Für Fortgeschrittene
Das GFA-BASIC ST Entwicklungssystem 3.0 Interpreter + Compiler DM 198,–

Und welches professionelle Werkzeug benutzt Frank Ostrowski, um den Compiler der Version 3.0 zu erstellen?

GFA-ASSEMBLER

Der GFA-ASSEMBLER ist ein leistungsfähiges Werkzeug für den Profi-Programmierer: Editor, Assembler und Linker in einem Programm, nachladbarer Debugger. Das Programm ist lauffähig auf allen ATARI ST, Monochrom- und Farbbildschirm. Und: Der GFA-ASSEMBLER ist schnell, sehr schnell. Die Bedienung ist einheitlich. Das Programm verfügt über eine eigene, schnelle Benutzer-Oberfläche. Durch die Integration von mehreren Programmteilen entfallen häufige Wartezeiten. Alle Programm-Module sind sowohl mit Maus als auch über Tastatur steuerbar. DM 149,–

GFA Systemtechnik GmbH
Heerdter Sandberg 30-32
D-4000 Düsseldorf 11
Telefon 0211/5504-0



Erhältlich in der Schweiz: DTZ DataTrade, Zürich